DTIC FILE COPY

COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY · STANFORD, CA 94305-2192

1

87-0242

OK. DTIC

AD-A221 875

CAPACITY ANALYSIS OF MULTIHOP PACKET RADIO NETWORKS UNDER A GENERAL CLASS OF CHANNEL ACCESS PROTOCOLS AND CAPTURE MODELS

José Manuel Rego Lourenço Brázio

Technical Report: CSL-TR-87-318

March 1987

This report is the author's Ph.D. dissertation which was completed under the advisorship of Professor Fouad A. Tobagi. This work was supported by the Defense Advanced Research Projects Agency under Contracts No. MDA 903-79-C-0201 and No. MDA 903-84-K-0249 and by a fellowship from the Instituto Nacional de Investigação Científica, Lisbon, Portugal.

[redacted]

COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY · STANFORD, CA 94305-2192

CSL-TR-87-318

José Manuel Rego Lourenço Brázio

Errata

Please replace the last two lines on page 181 with the following:

satisfies the above condition, one can choose $h(G_1, \cdots, G_L) = \sum_{r=1}^{L} G_i$, or any other functional in which the coefficients β_i are all nonzero. Some choices of $h(\cdot)$

Department of Electrical Engineering • Department of Computer Science

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 87-318	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CAPACITY ANALYSIS OF MULTIHOP PACKET RADIO NETWORKS UNDER A GENERAL CLASS OF CHANNEL ACCESS PROTOCOLS AND CAPTURE MODES		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT 6. PERFORMING ORG. REPORT NUMBER 87-318
7. AUTHOR(s) Jose Manuel Rego Lourenco Brazio		8. CONTRACT OR GRANT NUMBER(s) MDA 903-79-C-0201 MDA 903-84-K-0249
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford Electronics Laboratory Stanford University Stanford, CA 94305-2192		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Information Processing Techniques Office 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE March 1987 13. NUMBER OF PAGES 294
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Resident Representative Office of Naval Research Durand 165 Stanford University, Stanford, CA 94305-2192		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A packet radio network is a collection of geographically distributed packet radio units communicating over a shared broadcast channel. Usually not all radio units are within hearing range of each other, and thus multihop operation is required. These networks represent the natural extension of point-to-point packet-switched data networks when mobile operation is desired. An important difference exists, however, with respect to the latter: due to the multiaccess nature of the radio channel, the success of a packet at a destination depends

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

on the activity, during the transmission of the packet, of the neighbors of the destination, and on system parameters such as the type of signaling and received power levels. The conditions under which a packet is successfully received in the presence of interfering packets are designated as the *capture mode*. Due to the existence of multiuser interference, some form of coordination among the users is required when accessing the channel. This purpose is accomplished by the *channel access protocol*.

This thesis deals with the problem of the capacity analysis of a multihop packet radio network; namely, given a network specified by its topology, traffic pattern, channel access protocol, and capture mode, finding the maximum feasible link traffics compatible with the given traffic pattern. In this thesis we start by examining the capture behavior obtained from different signaling methods, and the question of the feasibility of implementation of different protocols under different signaling methods. The signaling schemes that form the basis of the discussion are narrowband and direct-sequence spread-spectrum. We then formulate a Markovian model that, through the appropriate setting of its parameters, allows the representation of the capture behavior of the different signaling methods, and the representation of the actions of the protocols in a general class that includes some of the main protocols of interest for packet radio applications. Examples of protocols in this class are Carrier Sense Multiple Access (CSMA), Busy Tone protocols, Disciplined-ALOHA, and ALOHA. From this model we derive throughput measures, and develop algorithms for finding the network capacity under a given traffic pattern. We then apply the analytical framework developed to the study of the relative performance of a number of channel access protocols in some simple topologies, and of the influence on this performance of system parameters such as the type of signaling, the bit durations and, for spread spectrum systems, the type of code assignment.

CAPACITY ANALYSIS OF MULTIHOP PACKET RADIO NETWORKS UNDER A GENERAL CLASS OF CHANNEL ACCESS PROTOCOLS AND CAPTURE MODELS

José Manuel Rego Lourenço Brázio

Technical Report: CSL-TR-87-318

March 1987

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-2191

Abstract

A packet radio network is a collection of geographically distributed packet radio units communicating over a shared broadcast channel. Usually not all radio units are within hearing range of each other, and thus multihop operation is required. These networks represent the natural extension of point-to-point packet-switched data networks when mobile operation is desired. An important difference exists, however, with respect to the latter: due to the multiaccess nature of the radio channel, the success of a packet at a destination depends on the activity, during the transmission of the packet, of the neighbors of the destination, and on system parameters such as the type of signaling and received power levels. The conditions under which a packet is successfully received in the presence of interfering packets are designated as the *capture mode.* Due to the existence of multiuser interference, some form of coordination among the users is required when accessing the channel. This purpose is accomplished by the *channel access protocol.*

This thesis deals with the problem of the capacity analysis of a multihop packet radio network; namely, given a network specified by its topology, traffic pattern, channel access protocol, and capture mode, finding the maximum feasible link traffics compatible with the given traffic pattern. In this thesis we start by examining the capture behavior obtained from different signaling methods, and the question of the feasibility of implementation of different protocols under different signaling methods. The signaling schemes that form the basis of the discussion are narrowband and direct-sequence spread-spectrum. We then formulate a Markovian model that, through the appropriate setting of its parameters, allows the representation of the capture behavior of the different signaling methods, and the representation of the actions of the protocols in a general class that includes some of the main protocols of interest for packet radio applications. Examples of protocols in this class are Carrier Sense Multiple Access (CSMA), Busy Tone protocols, Disciplined-ALOHA, and ALOHA. From this model we derive throughput measures, and develop algorithms for finding the network capacity under a given traffic pattern. We then apply the analytical framework developed to the study of the relative performance of a number of channel access

protocols in some simple topologies, and of the influence on this performance of system parameters such as the type of signaling, the bit durations and, for spread spectrum systems, the type of code assignment.

ACKNOWLEDGMENTS

Many people have contributed in many different forms to the taking to good term of the endeavor embodied in this dissertation. Foremost among those people I am indebted to is my family: my parents, who brought me up to be what I am, my wife, Barbara, who was an unwavering source of support and caring all through the sometimes very painful process of doing research and writing a dissertation, and my sons, Philip and Paul, who, in spite of not being given all the attention that was their prerogative to have received, always had spare cheerfulness to contribute to my days. To Prof. Fouad Tobagi goes my appreciation for his constant stimulation, constructive criticism, and demand for clear and rigorous thinking. To the excellent lectures of Prof. Donald Iglehart I owe some of the key concepts that I employed in the solution of the problem treated in this dissertation. To Ms. Doris Hsia, Jill Sigl, and Sandi Dodson I extend my recognition for their competence and dedication in the typing of parts of this and related work, and also for their administrative guidance during my stay at Stanford. I am also grateful to Willem Terluin, of Word Graphics, for his goodwill and competence in drafting the many figures that I produced during the course of this work. To the other students in my group goes a word of thanks for their supportive presence at all times. I would also like to acknowledge the many productive discussions with my officemate David Shur on the subject of packet radio.

I am deeply indebted to the many people at the Lisbon Institute of Technology (Instituto Superior Técnico) who at the beginning of my academic life guided, helped, and encouraged me. Special recognition is due to Prof. Fernando E. Rebelo Simões, who (quoting E. Çinlar) opened to me the doors of a life of randomness. To Profs. Manuel J. Abreu Faro and José M. Fonseca de Moura I would like to thank the support received for this endeavor. I also would like to gratefully acknowledge the support of the Luso-American Educational Commission, through which I received a Fulbright travel grant, and the Portuguese Research Council (Instituto Nacional de Investigação Científica), from whom I received a fellowship for most of the duration of this work.

CONTENTS

FIGURES

Chapter 1

INTRODUCTION

1.1 Packet Radio Networks: Overview

A *packet radio network* (PRN) consists of a collection of geographically distributed *packet radio units* (PRU's), communicating over a radio channel using packet switching techniques. In general not all PRU's are within radio hearing range of each other, and thus multihop store-and-forward operation is required. Packet radio networks combine the multihop nature of point-to-point store-and-forward networks with the multiaccess nature of broadcast networks such as some of those employed in local area packet communication.

Packet radio networks represent the natural evolution of point-to-point packet-switched data networks when mobile operation is desired, or when good wireline data links are not available. It is the latter reason that motivated the construction of the first operating packet radio network, the ALOHA system [Abra70]. The ALOHA system was a centralized system, in which a population of users sent packets to, and received packets from, a central station, with separate frequencies for

the inbound and outbound traffic. All traffic between any two users was routed through the central station. At the time the ALOHA system was developed, it was not technologically feasible to put into a unit small enough to be of practical interest for mobile operation, and in a cost-effective manner, the computing power needed for the processing and management functions required by packet-switched store-and-forward operation. It was during the 1970's that the advances achieved in the area of VLSI design and fabrication, in particular the development of microcomputers, allowed that goal to be attained. During this decade, the Defense Advanced Research Projects Agency (DARPA) of the Department of Defense of the United States initiated a research program whose purpose was to show the feasibility of the packet radio concept for mobile operation. An excellent overview of the organization and structure of the packet radio network contemplated in this program, as well as the goals of the program, can be found in [Kahn78].

The basic building block of a packet radio network is the PRU. Each PRU consists essentially of a radio section and of a digital section. The radio section contains the antenna, a transmit/receive switch, one transmitter, and one receiver ([Kahn78], [Fral75]). The digital section contains the elements necessary for performing the storing and routing of incoming packets, as well as controlling the operating parameters of the receiver and transmitter (Figure 1.1). Typically all communication in the network takes place using a single radio channel, in which case the operation of the transmitter and of the receiver is mutually exclusive.

Each PRU will in general act both as a repeater for transit packets, and as a local source and sink of digital information. Upon reception of a packet of which it is the intended destination, a PRU will either deliver it to a local user attached to it, or store it in an appropriate outgoing queue for later transmission on the radio channel. The transmission of the queued packets is attempted transmission

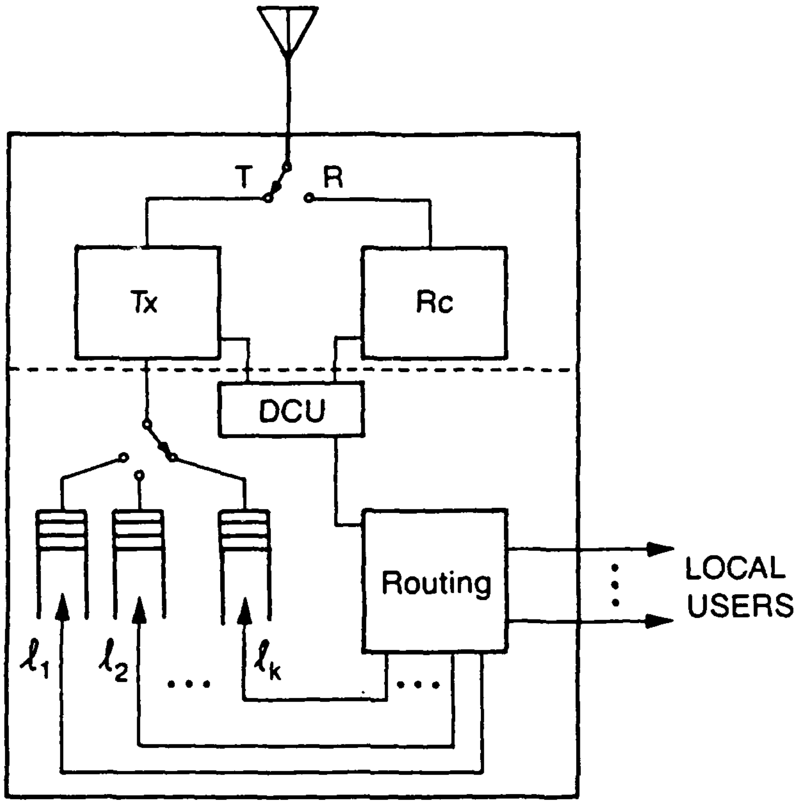


Fig. 1.1 Block structure of a packet radio unit

at times defined by a *scheduling algorithm*. At each of these times, an actual packet transmission will or will not take place, according to the decision made by the *channel access protocol* being used, whereby the decision may be based on the state of the network as perceived by the PRU. Once a packet is transmitted, it may or may not be successfully received at the intended destination, depending on the characteristics of the physical link and on the activity of the neighbors of the destination PRU. Indeed, in a packet radio network, and unlike the situation found in a point-to-point network, a packet sent from a transmitter to an intended immediate destination will be heard by PRU's other than the intended destination, having thus the capability to interfere with the reception of packets destined to these other PRU's (Figure 1.2). Depending on the specific situation considered, the

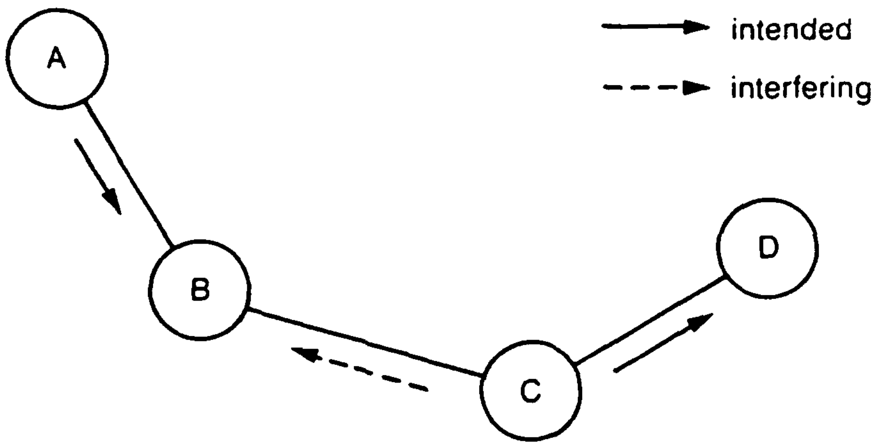


Fig. 1.2 Collisions in a packet radio network

overlap at a receiver of an intended packet and of interfering packets can prevent the correct reception of the former. The ability of a receiver to correctly receive a packet in the presence of interfering packets is referred to as *capture*. The specific conditions (dependent among other things on the signaling method used) under which a packet is correctly received in the presence of interference are designated as the *capture mode*. After the packet transmission takes place, and in the same way as in a point-to-point network, some acknowledgment mechanism is required to inform the source of a packet of the success or failure of the transmission.

We can look at a packet radio network as a set of *nodes* (the PRU's), interconnected by a set of directed radio *links* determined by the radio connectivity. The *throughput* of a link, the main performance measure of concern in this work, is defined as the fraction of time that the link is used for successful packet transmissions.

1.2 General Considerations on the Analysis Problem

Some of the aspects of the operation of a packet radio network described in the previous Section are difficult to model analytically, and cannot be included in a

general tractable analytical model. The main sources of difficulty in the analysis of the performance of a packet radio network are:

(i) *Store-and-forward operation:* It is difficult to keep track of the length of packets as they are forwarded from node to node in the network. This same difficulty had already been encountered in the delay analysis of point-to-point networks, in the context of packets whose lengths have an exponential distribution ([Klei64]). In this work it was concluded that, for most networks of interest, with moderate connectivity, the assumption that (*Kleinrock's independence assumption*) *the length of a packet is independently redrawn from the appropriate distribution each time the packed is received at a node within the network* has a negligible effect on the resulting average message delay. Whereas in general the exact solution of a point-to-point network would be a very difficult problem, due to the dependence between the interarrival times and the service requirements (the time between two consecutive packet arrivals can never be smaller than the length of the first of the two packets), with such an assumption a point-to-point network becomes a Jacksonian network of queues, admitting a very simple analytical solution.

(ii) *Retransmissions:* Due to collisions suffered at the intended destination, a packet at a node may have to be transmitted more than one time. Similarly to the situation in (i), it is difficult to keep track of the length of packets as they are retransmitted by a node. We would wish to make, for analytical tractability, an assumption similar to Kleinrock's independence assumption, i.e., that packet lengths are independently redrawn at each transmission attempt. Such an approximation would not be reasonable in a situation where the average time between the end of an unsuccessful transmission and the next retransmission is small, and the probability of packet loss is high, and

the variance of the packet length distribution is high, since then it would be very likely that a long packet would be transmitted soon again after a long packet finished transmission, contrary to what an independence-type assumption would predict. In a situation where the above circumstances do not exist, one would intuitively expect the above independence-type assumption to have a small impact on the predicted system performance. Such an assumption was used by Ferguson in [Ferg77] for the delay study of a single-hop unslotted ALOHA channel. In that study, it was stated that "simulation results indicated that this approximation was quite accurate."

(iii) *Queueing:* We can look at a packet radio network as a system of interfering queues, in the sense that the service time at a given queue, i.e., the time between the arrival of a packet at the head of a queue and its successful departure, is a function of the activity of neighboring queues. In particular, and as an example, one would expect in general such time to be larger at a queue if the neighboring queues are not empty, and thus very likely to be attempting to transmit packets, than if otherwise. The problem of interfering queues is a difficult one, and has been solved exactly only for very simple special cases ([Fayo79], [Nain85], [Sidi83a], [Sidi83b]). This situation, barring an unforeseeable breakthrough, is not expected to change substantially in the near future. Tractable models which give some insight into the performance of interfering queues can however be constructed, by disregarding some aspects of the problem. One possibility consists of assuming some form of decoupling between the states of different users ([Saad81], [Lee82]). Another possibility considers a heavy-traffic situation, in which all queues are always full. In this case the queueing aspects are unfortunately no longer represented, but the resulting model is nevertheless still useful to derive upper bounds on achievable throughputs i.e., capacity results. Still

another possibility is to consider packet buffers of size one, for which results on the packet service time (i.e., the time between the arrival of a new packet to the buffer and its successful departure) can be derived ([Ferg77]). In this case the queueing aspects are not fully represented, but one still obtains an indication of the delays experienced by the packets as they travel through the network.

(iv) *Propagation delay:* During the operation of a node, packets are considered for transmission at times dictated by a scheduling algorithm. At each of these times, the components of the state of the system that are locally accessible to the node are examined and, depending on the protocol in use, a transmission is attempted or not. Usually these components of the state information refer to the activity of packet radio units that the node in question can hear. The state information obtained by a node at a given time is a composite picture of the state of the network, obtained from delayed versions, according to the propagation times involved, of the states of the neighboring units. This delayed information may cause the system to enter states that would otherwise be forbidden under a zero propagation delay operation. Examples of these are states in which neighboring nodes transmit simultaneously under Carrier Sense Multiple Access. These states cannot exist in an ideal environment of zero propagation delay due to the action of the carrier sensing mechanism. Even in the simpler case where the propagation delays between all pairs of neighbors are taken to be equal, the modeling of nonzero propagation delay requires the storage of the history of the system over a period of time equal to the propagation delay. In the case of single-hop networks with an infinite population of users it is possible to obtain an analytical treatment of the situation of nonzero propagation delay, as long as the packet lengths are shorter than the propagation delay ([Klei75b]). In the

case of general multihop networks the need to store the history of the system for a nonzero time interval leads to models with a continuous state space, whose solution cannot be found in practice. Approximate models have been considered, where the time axis is divided into slots of duration equal to the propagation delay, and all transmissions start at the slot boundaries. With suitable assumptions, these models lead to discrete-time Markov chains, and can be solved by standard methods ([Toba77]).

1.3 Statement of the Problem

The problem studied in this thesis is that of determining the *capacity* of a network, as described in the following.

Let us consider a *heavy-traffic* situation, that is, a situation where each queue in the network has at any time an infinite number of packets to be transmitted. In this way, as soon as one packet is successfully transmitted, a new packet is immediately available to be scheduled for transmission. Consider also that we fix the channel access protocol, capture mode, and characteristics of the physical link, and let $\mathbf{P}$ be the vector of the remaining network operating parameters (e.g., average time between reschedulings, traffic generation rates, etc.). Let also $\mathbf{S}^*(\mathbf{P})$ be the resulting vector of heavy-traffic link throughputs. Consider now that, from a set of end-to-end traffic requirements and from a static routing procedure, a vector of required link throughputs $\mathbf{S} = \alpha\, \mathbf{S}^*(\mathbf{P})$, $0 \leq \alpha < 1$, results. One would intuitively expect such $\mathbf{S}$ to be feasible when the network is operating at operating point $\mathbf{P}$, in the sense that the network would be able to support the required link throughputs with finite queue sizes, and hence finite average message delays. Due to the analytical difficulties mentioned in Section 1.2, it has not been possible to prove this intuitive expectation

to be true, except in very simple special cases (see references given in Section 1.2). Nevertheless, the behavior described has been observed in simulation studies of a number of more complex network topologies and access protocols ([Toba85]).

Consider the set of feasible throughput vectors, or *feasible region*. According to the discussion above, this region can be defined as the set of values of $\mathbf{S}^*(\mathbf{P})$ obtained when $\mathbf{P}$ ranges over all possible values in the parameter space. From the point of view of the practical operation of the network, however, the operating points of interest are those lying on the boundary of the feasible region. Indeed, let $\mathbf{S}$ be a point in the interior of the feasible region, representing a set of required link throughputs, and let $\mathbf{P}_0$ and $\mathbf{P}_1$ be operating points such that $\mathbf{S}^*(\mathbf{P}_0)$ and $\mathbf{S}^*(\mathbf{P}_1)$ exist on the line joining the origin and the point $\mathbf{S}$, with $\mathbf{S}^*(\mathbf{P}_0)$ on the boundary of the feasible region and $\mathbf{S}^*(\mathbf{P}_1)$ *between* $\mathbf{S}$ and $\mathbf{S}^*(\mathbf{P}_0)$ (Figure 1.3). Again intuitively, one would expect the behavior of the system (say, in terms of average queue sizes) to be better at the operating point $\mathbf{P}_0$ than at $\mathbf{P}_1$, since in the former case the system is operating "farther away" from capacity. In addition, if only the *traffic pattern*, (i.e., the set of ratios between pairs of required link throughputs) is known, $\mathbf{P}_0$ is the natural choice of operating point, in the sense that it is the only value that will guarantee finite average delays for required link throughputs $\mathbf{S} = \alpha\,\mathbf{S}^*(\mathbf{P}_0)$, for all $0 \leq \alpha < 1$. Thus the determination of the points on the boundary of the feasible region is of special significance in the context of the network operation. We state now the problem to be addressed in this thesis:

> *Capacity problem:* Let a packet radio network, defined by its radio connectivity, channel access protocol, and capture mode, be given. Let also a desired traffic pattern be specified by a vector $\mathbf{S}$. We want to find the least upper bound of the set of values of α, where α is a positive real number, such that $\alpha\mathbf{S}$ is a feasible vector of link throughputs.

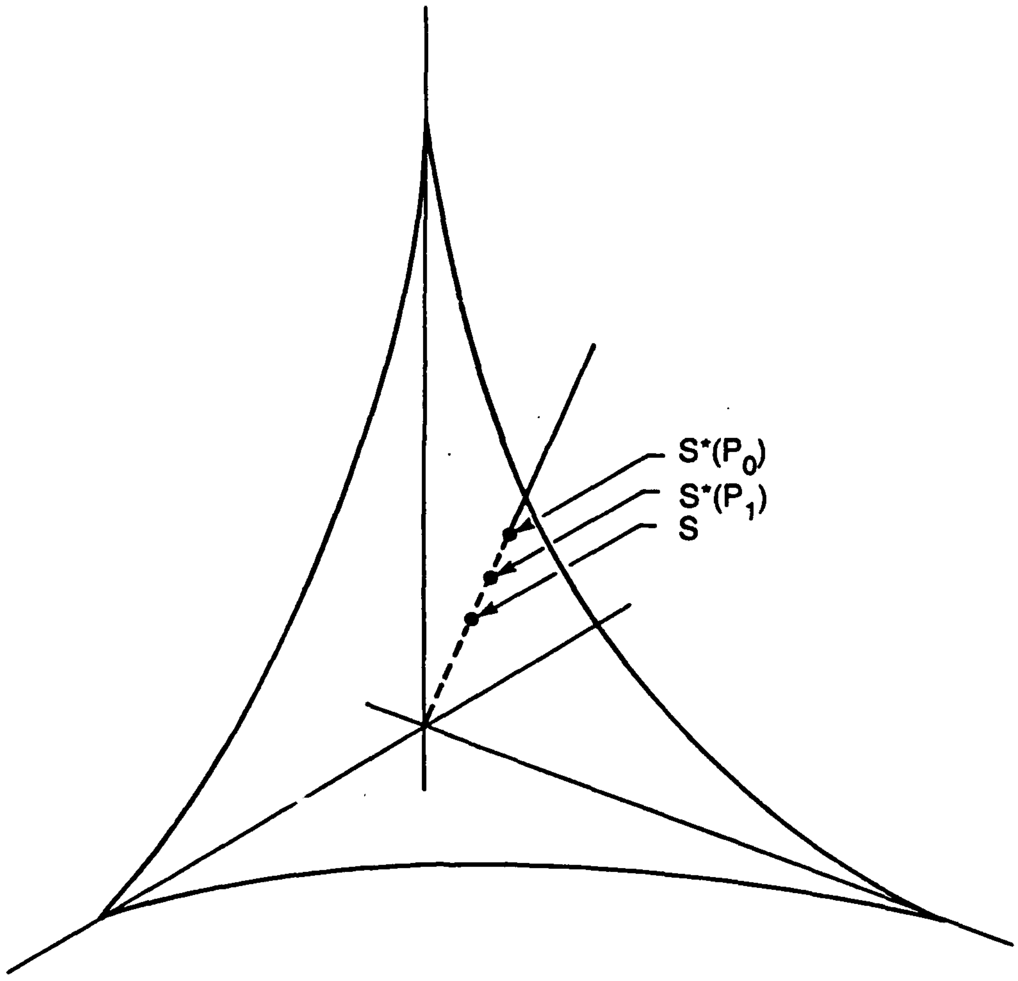


Fig. 1.3 Feasible region in S-space

Geometrically, the problem stated corresponds to finding the feasible point on the line passing through the origin and with the direction of $\mathbf{S}$, which lies farthest away from the origin. For its solution, the problem will be broken down into the following subproblems: (i) given an operating point $\mathbf{P}$, find the resulting heavy-traffic throughput $\mathbf{S}^*(\mathbf{P})$; (ii) given a set of link traffic requirements $\mathbf{S}_0$, find the operating point(s) $\mathbf{P}$ such that $\mathbf{S}_0 = \mathbf{S}^*(\mathbf{P})$; (iii) given a traffic pattern $\mathbf{S}_0$, find the maximum $\alpha > 0$ such that $\alpha\, \mathbf{S}_0$ is a feasible set of link traffic requirements, and the corresponding operating point.

We put as a goal of the model to be developed that it be applicable to the analysis of a large class of access protocols, including those commonly considered for packet radio operation (ALOHA, CSMA, Busy Tone protocols), and of the types of capture resulting from the narrowband and spread-spectrum signaling schemes commonly used. From such a model we shall later be able to make a comparative study of the influence of those variables on the performance of the packet radio system.

1.4 Survey of Related Work

We give in this section a short survey of work in the area of performance analysis of packet radio networks. Our goal is not to be exhaustive, but rather to present a number of papers that cover the relevant aspects of the problem. We consider separately the analysis of single-hop and multi-hop systems.

Literally speaking, a single-hop network is one where packets only need to travel one hop to reach their intended destination, and thus where no store-and-forwarding is necessary. According to this definition, having the single-hop property would depend not only on the topology but also on the traffic pattern involved, in such a way

that, for example, a ring network with $N > 3$ nodes and only nearest-neighbor end-to-end traffic would be classified as single-hop. However, from the point of view of the models required for capacity analysis, the nature of the end-to-end traffic pattern is relatively unimportant, whereas the topology is one of the determining factors. The distinction between single-hop and multihop seems to reside rather in whether or not there exists dependence between the actions of PRU's that are not within range of each other, "carried" by the intermediate PRU's. Indeed channel access protocols introduce in general dependencies between the activity of neighboring PRU's, and these dependencies "propagate" across the network. The literal definition given above does not reflect, then, the established meaning of the term *single-hop*. To be consistent with this meaning, we shall accept that a single-hop network consists either of (i) a population of users such that any pair in the population is in radio connectivity, or (ii) a centralized (star) network, with a receiver in the center of the star, and a population of transmitters in the outer positions of the star. Case (i) is referred to as a *fully-connected* environment, and case (ii) as a *hidden-terminal* environment. In this latter case, individual transmitters may be in radio connectivity with a subset of the set of transmitters.

1.4.1 Single-hop systems

The first random-access protocol considered and analyzed was the ALOHA protocol. Under this protocol, a new message is transmitted as soon as it arrives at a PRU, and is retransmitted after a random rescheduling interval if a previous transmission of it resulted in a collision. Some acknowledgment mechanism is assumed whereby the transmitter learns about the success or failure of the transmission, and thus of the need for a retransmission. The ALOHA protocol is called *slotted* if the packet transmissions are synchronized so that their starting times are restricted to

occur at multiples of a given *slot size*, and *unslotted*, otherwise. In [Abra77], Abramson gives a collection of results concerning the throughput analysis of centralized ALOHA systems, for a number of distinct traffic situations and both the slotted and unslotted cases. Most of the results given concern the situation designated as *zero capture*, in which any time overlap between two packets at a receiver leads to the destruction of both packets. Some results are also given for *power capture*. In this situation, a packet transmission with received power P_D is successfully received if the interfering packet has a received power P_I such that $P_I < \beta P_D$, where β is a parameter between zero and one, with β^{-1} being designated the *capture ratio* ([Robe72]). The network considered consists of a population of traffic generators distributed on the infinite plane with a given density. One of the conclusions derived for these systems is that, when the desired traffic (i.e., throughput) density is uniform, there is a maximum radius (the Sisyphus distance) beyond which communication with the central receiver is not possible. In [Ferg77], Ferguson provides an approximate delay analysis of a centralized unslotted ALOHA system with a finite number of users, each with a single-packet buffer. He studies the average message delay as a function of the mean rescheduling delay and the number of users, for different packet length distributions. He also compares the results of the analysis with simulation results. In [Klei75a], Kleinrock and Lam analyze the case where a central station receives packets from a finite population of transmitters, each possessing a single-packet buffer, using slotted ALOHA. They show that under certain conditions the operation of the channel becomes unstable, leading to a situation of low throughput in which most users have backlogged packets to be transmitted, the retransmission attempts of which keep giving rise to new collisions. In the companion paper [Lam75], dynamic control procedures are introduced and analyzed that prevent this behavior from occurring. Along a separate direction, in [Saad81], Saadawi and Ephremides provide an approximate queueing analysis of a centralized

slotted ALOHA system with a finite population of users, each of which possesses a buffer of infinite size, thus obtaining delay and stability results. The authors achieve analytical tractability by representing the system by a two-level (user level and system level) description, and assuming decoupling between the two.

The results of the previous papers show that ALOHA achieves a relatively low channel utilization, due to the uncoordinated attempts of use of the channel by the user population. Its performance can be improved in a fully-connected environment if terminals listen to the channel just prior to transmitting, and refrain from transmitting if they sense the channel busy. The protocol resulting from this mode of operation is designated as Carrier Sense Multiple Access (CSMA). This protocol was formally introduced and analyzed for its throughput-delay performance by Kleinrock and Tobagi in [Klei75b]. The situation analyzed consisted of a fully-connected infinite population of terminals communicating with a central station. Their results showed that the carrier sensing mechanism introduced a substantial improvement in the system performance, as compared with that of ALOHA. However, the carrier sensing mechanism does not prevent collisions if some of the terminals cannot sense the activity of other terminals (hidden terminals), as in a star network, for example. In [Toba75], Tobagi and Kleinrock studied a single-hop hidden-terminal network under CSMA. Their results showed that the existence of hidden terminals introduces a severe degradation on the system performance relative to that of the fully-connected situation. In order to obviate this shortcoming, they proposed the Busy-Tone Multiple Access (BTMA) protocol. According to this protocol, the central receiver, which is in radio connectivity with all transmitters, sets up a "busy tone" on a separate channel whenever it detects any transmission, and the transmitters refrain from transmitting whenever a busy tone is sensed on the busy tone channel, or any activity is detected on the data channel. In this way the occurrence of simultaneous transmission attempts is very much reduced (and even eliminated,

in a situation of zero propagation delay), and the performance improved.

All of the above papers assume a simple form of capture, mostly zero capture, characteristic of narrowband signaling schemes. Different types of capture are exhibited by another class of signaling schemes, *spread-spectrum* signaling. In one such capture mode, *time capture*, the first packet to reach an intended receiver can usually be correctly received if its arrival is separated at least by a "capture time" from the arrival of the next packet. In [Davi80], Davis and Gronemeyer present the analysis of a centralized finite-population slotted-ALOHA system with time capture, in which each user has a single-packet buffer. In this system, users can be at different distances from the central receiver. If every user transmitted at the slot boundaries, the packets would arrive at the central receiver in order of increasing link range, thus leading to discrimination against transmitters which are sufficiently far from the receiver. In order to avoid this situation, the authors consider that terminals delay their transmissions for a period of time which is the sum of a quantity linearly dependent upon their distance to the receiver and a uniformly distributed random variable, in such a way that the arrival times of the packets at the central receiver are uniformly distributed in the same time interval. The authors then show that such system possesses excellent delay and throughput performance, and that by careful selection of the system parameters system stability can be maintained even in the presence of severe fluctuations in user population or traffic loading.

The above papers cover the main aspects of the multiaccess problem in a single-hop environment. A good general reference for the earlier work is contained in Chapter 5 of [Klei76].

Still within the framework of single-hop networks, other authors consider some more elaborate capture and channel models. In [Rayc81], Raychauduri considers a single-hop network with a finite population of transmitters, using a *Code Division*

Multiple Access (CDMA) scheme. In this scheme, each transmitter uses a different code in the encoding of the data in their packets. A pool of separate receivers, each of which can "tune in" to a different code, is assumed, and a probability of correct packet reception dependent on the level of multiuser interference is considered. For this model, throughput-delay results are derived. In [Stor84], [Toba84],and [Stor85], Storey and Tobagi consider a more complex model, in which the full-duplex operation of the PRU's is represented, and where the effect of coding of the data, prior to modulation, on the system throughput is studied. The former of these papers also considers a "channel load sense access protocol," in which radios are blocked from transmitting when the channel is heavily loaded. This study considers both the cases of zero and non-zero propagation delay. In [Purs83], Pursley considers asynchronous frequency hopping systems with fixed length packets, to find the throughput and the probability of a packet being correctly received.

Aside from these papers, which focus on the multiaccess aspect of the problem, there is a vast literature just devoted to the characterization of multiuser interference in spread-spectrum systems. The purpose of these studies is usually to compute the probability of bit errors as a function of the type of modulation and signaling, the codes employed, and the (fixed) number of overlapping signals. A good source of references on the subject is the May 1982 Special Issue on Spread-Spectrum Systems of the IEEE Transactions on Communications.

1.4.2 Multihop Systems

One of the first studies on multihop networks was done by Gitman in [Gitm75]. The system considered consisted of a set of clusters (each with an infinite population of users) that communicate with a central station. The clusters are not within hearing range of the station, and thus a set of repeaters is employed, each servicing

one cluster. The clusters cannot hear each other, as well as repeaters other than the ones assigned to relay their messages, but each repeater can hear, and be interfered by, a given number I (taken as a parameter) of other repeaters. The slotted ALOHA protocol is used. The system operates using different frequencies for the inbound and outbound traffic. The paper analyzes the *capacity* of such a system, and discusses techniques for performance improvement. The techniques considered are the use of directional antennas at the central station or at the repeaters, and the use of multiple transmitters at the central station. The author analyzes the bottlenecks in the system, and discusses the range of system parameters for which each of these techniques can increase the system throughput. In [Toba80a] and [Toba80b], Tobagi considered a similar system, but with inbound traffic only. He considered two different types of connectivity for the repeaters: *fully connected*, and *star connected*. He assumed that repeaters had a finite storage capacity, and studied the throughput-delay performance of such system for both the slotted ALOHA and CSMA protocols. Two different retransmission strategies at the repeaters were considered: (i) *immediate first transmission* (IFT), whereby a packet just arrived at the head of the repeater queue is attempted transmission in the next slot with probability one, and (ii) *delayed first transmission* (DFT), whereby such a packet incurs a geometrically distributed delay upon its arrival at the head of the queue. The paper also studied the improvement in performance due to increasing the buffer size M. The main conclusion drawn from this study is that there is a slight increase in performance from $M = 1$ to $M = 2$, but that for $M > 2$ no significant improvement in performance is obtained relative to the case $M = 2$, and thus the system is *channel-bound* rather than *storage-bound* for the higher values of M. In [Yemi80], Yemini analyzed a tandem (chain) network for two variants of slotted ALOHA, and derived the capacity of the resulting system.

A number of other studies have looked at networks where nodes are considered

to be randomly placed on the plane, with the connectivity being determined by a "transmission radius." Kleinrock and Silvester, in [Klei78], and Silvester, in [Silv80], consider the problem of finding the transmission radius that maximizes the system throughput, and conclude that such a radius should be chosen to be such that, on the average, each PRU can be heard by approximately six other PRU's. The former of these papers introduces the notion of *expected forward progress,* defined as the average reduction in distance towards the destination of the packet achieved in one hop. In [Taka84], Takagi and Kleinrock consider a similar situation, but in which power capture exists, for both slotted ALOHA and CSMA. The system performance measures used are the *expected forward progress* and the *probability of success* of a transmission. It is observed that for a slotted ALOHA system, perfect capture offers an improvement over zero capture of 36 percent on the expected progress. It is also remarked that the improvement obtained by CSMA over ALOHA on the expected progress amounts only to 6 percent, which is much less than that obtained in a fully-connected environment. This fact is attributed to the existence of hidden terminals. In [Nels84], Nelson and Kleinrock consider a similar network, and determine throughput equations for slotted ALOHA with power capture. Their results show that increasing the capture parameter increases the throughput of the network, and also show the tradeoff between the probability of successful transmission and the expected forward progress. In [Hou86], Hou and Li consider three transmission strategies: (i) *most forward with fixed radius*, in which a node will transmit with a fixed transmission radius R to a neighbor chosen such that the largest forward progress results, (ii) *nearest with forward progress*, in which the transmission power is adjusted to be just strong enough to reach the nearest neighbor which will result in forward progress, and (iii) *most forward with variable radius*, in which the transmission radius is adjusted to reach the destination receiver. They study throughput and forward progress for such a system, and show that the network

can achieve higher throughput and average progress by adjusting the transmission power. Among the three transmission strategies considered, (iii) is seen to lead to higher throughput and progress, with the performance being maintained even with a changing network topology.

In [Sous85], Sousa and Silvester consider the same problem, but taking a different approach regarding the connectivity and capture model. They remark that, in reality, a "hard" transmission radius does not exist, but rather the successful reception of a packet is a function of the relative strengths of the desired signal and of the interference at the receiver. They then consider a spread-spectrum system for which they develop a multiuser interference model based on an inverse power propagation law. For this system it is shown that the optimum addressing range should be chosen so that, on the average, there are $1.3\sqrt{K}$ terminals closer to the transmitter than the receiver, where K is the effective maximum number of simultaneously successful transmissions.

A number of other works have addressed performance issues in general multi-hop topologies. In [Nels85a], Nelson and Kleinrock introduce a variant of CSMA called *rude-CSMA*, under which a node transmits sometimes even if the channel is sensed busy, and present a model for the analysis of arbitrary topologies under this protocol Using this model, they analyze (i) a six-node grid network, and (ii) one randomly-generated seven node network with a variant of CSMA called *rude-CSMA*. Their results show that, for case (i), the optimal strategy consists of transmitting with a nonzero rate even if the channel is sensed busy, while in case (ii) the optimal strategy is to act as in CSMA. In [Taka85], Takagi and Kleinrock analyze two multi-hop topologies, operating under slotted-ALOHA, in which a number of end users exchange packets via a number of intermediate repeaters. Each repeater possesses a single-packet buffer. A detailed Markovian state description

(*à la* networks of queues) is constructed, and the throughput-delay performance is analyzed. The paper then proposes different ways of reducing the average delay for a given throughput: (i) *transmission suppression*, in which a repeater with no buffer space available transmits a "busy tone" in a separate channel to its neighbors to indicate that they should not attempt to transmit a packet to it in the next slot, (ii) *transmission acceleration*, in which a node transmits a packet in the next slot with probability one, instead of same probability $p < 1$, if the neighbors of the destination are known to have empty buffers in the current slot (the capability to acquire this information is assumed), and (iii) *multiple buffers for repeaters*. The techniques of (i), and (i) and (ii) combined are seen to lead to a significant performance improvement. The technique of (iii) has effects similar to those reported in [Toba80a] and [Toba80b], namely that there is a marked improvement when the repeater buffer size is increased from one to two, but little improvement after that. In [Lee82], Lee and Silvester introduce a model for the approximate queueing analysis of multihop networks with general topologies under slotted ALOHA. The model replaces the detailed description of the interference at a node due to the activity of neighboring nodes by an average interference parameter, identically and independently distributed from slot to slot. In this way, decoupling between the activity of different queues is obtained. The paper derives expressions for the generating functions of the distribution of queue sizes, and obtains delay and throughput equations. The results are then compared, for some sample networks, to results derived via simulation, with good general agreement. In [Boor80], Boorstyn and Kershenbaum introduce a heavy-traffic Markovian model for the analysis of CSMA in arbitrary symmetric topologies with exponentially distributed packet lengths. They show that the steady-state distribution of the stochastic process describing the activity of the transmitters has a product form, and derive throughput equations for the case of perfect time capture. The same authors together with Sahin analyze

in [Boor82] the same system when echo acknowledgments are considered. Maglaris, Boorstyn and Kershenbaum present in [Magl83] an extension of the model to Coxian packet length distributions. Extensions to other protocols and capture modes, as well as a formal analysis of similar systems, are given by Tobagi and Brázio in [Toba83], [Braz84], [Braz85a], and [Braz85b]. In [Chen85], Chen and Boorstyn give an approximate analysis of CDMA networks, and in [deSo85] deSouza, Chen and Boorstyn present a comparative evaluation of a number of access protocols, as well as the effects on performance of hearing radius. Starting from the same basic model, Yemini presents in [Yemi83] a macroscopic model, applicable to large user populations, based on ideas from Statistical Thermodynamics. In this model, an analogy is made between network quantities, such as concurrency and average rate of blocking, to macroscopic quantities in a large system of particles such as a gas. Results available for such systems, such as those concerning critical phase transitions, are translated into corresponding results concerning large networks.

1.5 Outline of this Dissertation

We now describe the contents of this dissertation. Chapter 2 deals with signaling and capture. This Chapter considers the two signaling methods employed in packet radio communication, namely narrowband and spread-spectrum, and characterizes qualitatively the capture properties obtainable therefrom. Section 2.1 defines narrowband signaling, and describes the *power capture* typical of this type of signaling. Section 2.2 is devoted to spread-spectrum signaling. It presents the general properties of spread-spectrum signaling, and briefly describes the generation of the signal for the *direct-sequence* and *frequency-hopping* methods of spread-spectrum signal generation. It then discusses the different choices available for the assignment of

code sequences to different users and to different data bits belonging to a given user, and the impact of such choices on the synchronization and bit error properties of the corresponding systems.

Chapter 3 deals with channel access protocols. Section 3.1 examines the kind of information required by a channel access protocol. Section 3.2 presents a number of definitions relevant for the specification of channel access protocols. Section 3.3 classifies the protocols commonly considered for packet radio operation in three classes, according to the nature of the information required for their operation: (i) ALOHA protocols, (ii) activity sensing protocols, and (iii) Busy Tone protocols. It then defines the main protocols of practical interest in each of these classes.

Chapter 4 describes the analytical model considered in this work. Section 4.1 defines formally the elements that specify a packet radio network. Subsection 4.1.1 deals with the topology and traffic requirements. Subsection 4.1.2 defines the class of channel access protocols considered for analysis, and gives their formal specification in terms of the *blocking* between radio links, i.e., the specification of which transmissions are allowed to start at a given time, given the set of transmissions already taking place at that time. The class of protocols considered includes most of the protocols used in packet radio applications. It further defines a subclass of the above class of channel access protocols, for which the protocol decisions are based only on the state of the network transmitters, and whose elements lead to a form of decoupling in the representation of the network activity and in the evaluation of the network throughput. Subsection 4.1.3 describes the general capture model considered for analysis, and defines two particular capture modes, *zero capture* and *idealistic perfect capture*, that will be later considered in a number of examples. Finally, Section 4.2 introduces the network operating model that is considered in the remaining part of this work, and discusses the validity of its assumptions for

the purpose of capacity analysis.

Chapter 5 presents the analysis of the class of protocols for which the protocol decisions are completely specified by the state of the network transmitters. The protocols in this class lead to decoupling between the description of the activity of the receivers and the activity of the transmitters, and the corresponding systems are referred to as *decoupled systems.* Section 5.1 presents a Markovian model for the description of the transmitter activity. The model is a direct extension of a model introduced by Boorstyn and Kershenbaum in [Boor80] for the analysis of Carrier Sense Multiple Access. This Section characterizes the state space of the corresponding stochastic process, derives the balance equations for its steady-state probability distribution, and investigates the conditions under which this distribution possesses a product form solution. The existence of a product form solution is shown to be equivalent to symmetry in link blocking. This Section also shows that the product form solution is computationally NP-hard. Section 5.2 presents a Markovian model for the description of the activity of the receivers, and derives the equilibrium equations of the corresponding stochastic process in a form that emphasizes the decoupling afforded by the class of protocols considered in this Chapter. Section 5.3 presents the derivation of throughput measures from probability measures and hitting times associated with the processes describing the activity of the receivers and of the transmitters. This Section also presents the derivation of the throughput equations for a four-node ring network, as an example of application of the formalism developed in this Chapter.

Chapter 6 presents the analysis of the more general class of channel access protocols that do *not lead to decoupling* between the descriptions of the activity of the transmitters and the activity of the receivers. Section 6.1 presents a Markovian model for the description of the network activity. It presents the characterization

of the state space of the associated stochastic process, and presents the balance equations satisfied by its steady-state probability distribution. Section 6.2 presents the derivation of throughput equations, using an approach similar to that of Chapter 5.

Chapter 7 extends the results of Section 5.1 on the existence of a product form solution for the process describing the transmitter activity, to the case of nonexponential packet length distributions and non-Poisson scheduling processes. The resulting non-Markovian processes belong to a class known as Generalized Semi-Markov Processes (GSMP's). Section 7.1 introduces the problem. Section 7.2 gives the definition of a GSMP, and presents the properties of a GSMP that are of interest for our applications. Section 7.3 introduces two constructions for the rescheduling point process, designated as *Continued Renewal Rescheduling* and *Restarted Renewal Rescheduling,* that reduce to a Poisson process for exponential rescheduling intervals, and formulates the resulting transmitter activity processes as GSMPs. Section 7.4 studies the existence of a product form solution under each of the rescheduling mechanisms considered. For *Continuous Renewal Rescheduling* with Poisson rescheduling processes and for *Restarted Renewal Rescheduling* with arbitrary rescheduling processes these conditions are found to be identical to those found in Chapter 5 for the case of exponential packet lengths. *Continuous Renewal Rescheduling* with arbitrary rescheduling intervals is found not to possess in general a product form solution. Section 7.5 formulates the transmitter activity process as the state of a queue with state-dependent arrivals, thus establishing a connection between the processes considered in this work and some processes considered in the queueing literature.

Chapter 8 deals with the computational aspects of the capacity analysis. Section 8.1 intrcduces the problem. Section 8.2 is devoted to the computation of the link

throughputs given the channel access protocol, the capture mode, and the operating parameters (rescheduling rates and average message lengths) of the network. Subsection 8.2.1 considers the more general protocols of Chapter 6, which do not lead to decoupling between the activity of the transmitters and the activity of the receivers. These protocols require the numerical setting up and solution of the balance equations of the processes involved. The enumeration of the state space of these processes is reduced to the traversal of a directed graph, and a breadth-first search (bfs) is considered for this purpose. A number of properties of the bfs enumeration of the state space are then established. From these properties, a characterization of the order in which states are enumerated and an efficient algorithm for that enumeration are derived. Finally, algorithms and data structures appropriate for a computer-implemented enumeration of the state space, setting up of the balance equations, solution of the systems of linear systems involved, and computation of throughput, are described. Subsection 8.2.2 considers the restricted class of protocols considered in Chapter 5, which lead to decoupling between the activity of the transmitters and of the receivers. Its contents parallel those of Subsection 8.2.1. Section 8.3 considers the problem of, given a set of link throughput requirements, solving for the network operating parameters that attain those requirements. This Section presents a fixed-point iteration algorithm for the solution of the problem. Section 8.4 considers the problem of finding the capacity corresponding to an a *priori* given traffic pattern. Two strategies are presented: (i) a trial-and-error binary search method, at each step of which the feasibility of a tentative value of the capacity is tested, and (ii) a parametric method that, given an arbitrary linear functional h of the rescheduling rates, finds the network throughput as a function of the value assigned to h, and then obtains the capacity by maximization over that value.

Chapter 9 presents numerical applications. This Chapter gives capacity results

for a number of parametric topologies (rings, chains, and stars) and randomly generated topologies. A number of Busy Tone protocols, the Carrier Sense Multiple Access protocol, and two ALOHA protocols are considered, in both narrowband and spread spectrum environments. Section 9.1 specifies the spread spectrum and the narrowband systems considered. It also describes the noise and the capture model, the topologies, traffic patterns, and access protocols considered. Section 9.2 presents the numerical results. Even though only small networks can be accommodated due to the fast growing computational complexity of the analysis, these results allow some insight to be gained. In particular, they establish a ranking of the protocols in terms of their capacity performance, and illustrate the tradeoff achieved by the different protocols and signaling methods in controlling collisions at the receivers versus allowing the transmitters wide access to the channel.

Chapter 10 concludes with some general remarks on the problem of performance evaluation of packet radio networks, and with some suggestions for future research.

Chapter 2

SIGNALING AND CAPTURE

This chapter is concerned with one of the factors that influence the correct reception of a packet by an intended receiver in the presence of multiuser interference, the signaling method. The signaling methods examined are *narrowband* and *spread-spectrum*, together with the capture behavior obtained therefrom. Section 2.1 defines narrowband signaling, and describes the *power capture* typical of this type of signaling. Section 2.2 is devoted to spread-spectrum signaling. It presents the general properties of spread-spectrum signaling, and briefly describes the generation of the signal for the *direct-sequence* and *frequency-hopping* methods of spread-spectrum signal generation. It then discusses the different choices available for the assignment of code sequences to different users and to different data bits belonging to a given user, and the impact of such choices on the synchronization and bit error properties of the corresponding systems. The discussion of this Chapter is complemented by a quantitative model for the analysis of multiuser interference presented in Appendix I.

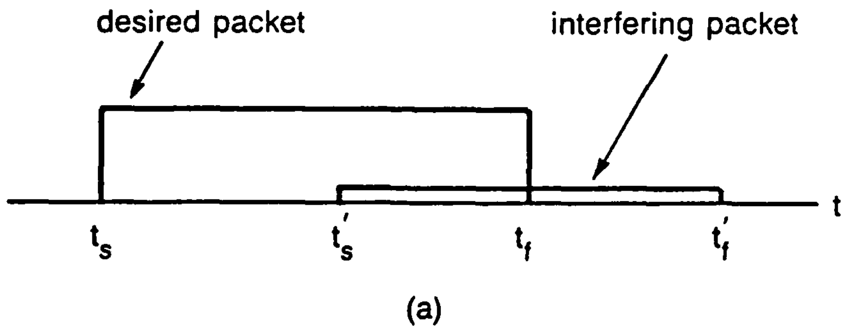
2.1 Narrowband Signaling

In narrowband signaling, the binary sequence of pulses representing the data modulates a carrier directly. Examples of modulation types commonly used for narrowband data transmission are amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), as well as variants of these ([Shan79]). All these schemes possess the properties that

(i) the presence of a transmission using the modulation scheme in question is easily detected;

(ii) an overlap in time at a receiver with a signal with the same modulation type and of comparable power will give rise to bit errors in the desired data.

Narrowband schemes exhibit *power capture:* if P_D is the received power of the desired signal and P_I is the received power of the *interfering signal*, a desired data bit will be correctly received if $P_I < \beta P_D$, for some $0 \leq \beta \leq 1$, with β^{-1} being called the *capture ratio.* The case $\beta = 0$ is called *zero capture* (the stronger signal is always received with error as long as there is any overlap), and the case $\beta = 1$ is called *perfect capture* (the stronger signal is always correctly received). (Of course, this is a simplified (deterministic) model, that does not take into account the probabilistic nature of the situations involved.) Even though individual bits of a packet may be correctly received in the presence of overlaps with other weaker signals, the correct reception of a packet depends on additional characteristics of the receiver. Consider the situation shown in Figure 2.1a, in which a desired packet arrives at, and starts being successfully received by, a receiver whose channel is idle. This packet is later overlapped by an interfering packet of amplitude small enough not to cause bit errors, and whose end extends beyond the end of the first packet. If, by monitoring the sudden decrease in the power of the received signal, or by examining



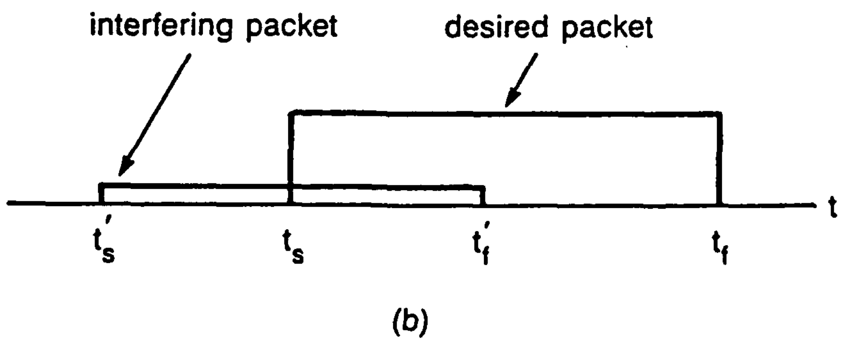


Fig. 2.1 A desired signal overlapped by a weaker signal

a possible byte count field in the packet header, the receiver can determine the time at which the first packet ends, then its reception will be successful. Otherwise, the receiver will assume the data bits of the remaining of the second packet to belong to the first one, thus acquiring incorrect data. Similarly, in the situation shown in Figure 2.1b, where the interfering packet arrives first, if the receiver can detect the increase in the received power due to the start of the desired packet and has enough "intelligence" to reset at that point the packet processing functions, then the desired packet will be successfully received. Otherwise, it will be lost. In a simplified model of power capture, we can assume that the correct reception of the desired packet is determined by the satisfaction of the condition $P_I < \beta P_D$ (see [Robe72], where it is assumed that a desired packet overlapped by an arbitrary number of interfering packets will be successfully received if each interfering packet individually satisfies this condition).

Random fluctuations due to receiver thermal noise are superimposed on the signal, and can also give rise to bit errors. The analysis of the noise performance of narrowband signaling schemes can be found in most elementary communications textbooks (e.g., [Shan79]).

2.2 Spread-Spectrum Signaling

The essential idea behind spread-spectrum is that of encoding the data bits in the transmitted signal prior to modulating a carrier. Quoting from [Pick82],

> "Spread spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery."

According to this definition, the coding aspect is essential. Modulation schemes such as FM, which occupy a bandwidth in excess of the strict minimum necessary, are not considered as spread-spectrum.

General properties of the spread-spectrum signal are:

(i) it is difficult for a receiver, without the knowledge of the codes being used in a transmission, to detect its presence;

(ii) spread-spectrum offers a larger degree of immunity than narrowband to bit errors caused by overlaps with similar signals; these similar signals can be, in particular, time delayed versions of the desired signal, whose presence is due to multiple propagation paths between the transmitter and the receiver (multipath components).

We examine in the following sections the generation of the spread-spectrum signal, as well as some system aspects such as synchronization and code assignment.

2.2.1 Signal Generation

One way of achieving Spread Spectrum operation consists of the use of *direct sequence* (DS) modulation. Assume that the data is represented as a sequence of binary digits. A bit of information (say 0) is encoded as a well specified sequence of ± 1 pulses, or "chips". We call this sequence the *code waveform.* Bit 1 will be normally represented by the complementary sequence. It is the sequence of chips resulting from the concatenation of the code waveforms corresponding to the bits in the data stream which then modulates a carrier (usually using some variant of phase modulation [Dixo84]) and is transmitted. The transmitted chip sequence can thus be viewed as the result of multiplying, for each bit, the corresponding code waveform by $+1$, if a zero is to be transmitted, or by -1, if a one is to be transmitted, and

transmitting the resulting waveforms in succession. The sequence of codes used in this process is referred to as the *code sequence*. In order to perform the reception of the data, the receiver is equipped with filters matched to the code waveforms used to represent the data bits. These filters perform the crosscorrelation between the input signal and the waveform whose presence the filter is intended to detect. As the incoming signal is processed, whenever there is a match the response of the filter will exhibit a peak, ideally narrow in width and high in amplitude, corresponding to the main lobe of the autocorrelation of the signal. For the scheme above indicated, a data bit zero will give rise to a positive pulse and a data bit one will give rise to a negative pulse. In addition to the detection of the presence and value of the data, these peaks are used for the acquisition of bit timing. In the absence of a matching codeword, the response is of much smaller amplitude.

Another spread spectrum technique is *frequency hopping* (FH)(see for example [Kahn78]). One method of generating a FH signal can be viewed as a modification of frequency shift keying. In FSK, a local oscillator with a given (fixed) frequency is frequency modulated by the data sequence. An instance of a frequency hopping signal can be obtained from a variant of this scheme by having instead the data sequence frequency modulate the output of an oscillator, whose frequency is changed at regular time intervals (the *dwell time*) according to a pseudo-random pattern known to both the transmitter and receiver. Here, the codes are the sequences of frequencies that are selected to be frequency modulated by the data signal. In order to receive the original data, the receiver multiplies the received signal by a synchronized replica of the output of the oscillator used to generate the signal, thus obtaining an FSK signal modulated by the original data sequence. We shall not consider FH in the remainder of this work, although the formalism to be presented is applicable to its analysis.

2.2.2 Synchronization

Another important aspect of the definition of spread spectrum presented above is that of *synchronization:* in order to be able to receive the data correctly, a receiver must not only know what code to use, but also the starting point in time of the code sequence. A common solution is to precede each packet transmission by a *preamble*, the bits of which are encoded using a fixed code waveform chosen to have good correlation properties (i.e., such that its *periodic autocorrelation* exhibits a narrow main lobe and sidelobes with much smaller amplitude, and its crosscorrelation with the other codes used in the system has small maximum values). The presence and time origin of the preamble code waveforms can then be detected, for example, by means of a matched filter, or of a sliding correlator([Dixo84]). The repetition of the code waveform will give rise to a series of pulses that indicate the position of the boundaries of the data bits of the received packet. The preamble may also serve to convey additional control information, such as the type of error correction and, in the case where not all users use the same code sequences, the characteristics of the code waveforms used in the transmission of the data portion of the packet. Having successfully processed the preamble of an incoming packet, the receiver at a node will thus lock onto the data portion of the packet until its transmission is completed (unless of course, in the meanwhile the transmitter was allowed to become active, thus aborting the reception, due to the half-duplex operation of the PRU's). Errors in the processing of the preamble may be due to overlaps in time with another packet's preamble (*preamble collision*), or due to the presence of other ongoing transmissions, which act as background noise and thus reduce the effective signal-to-noise ratio. The probability of such errors depends on the set of transmitters which are active at the start of the preamble, on the evolution of the system during its transmission, and on the codes of the interfering signals. These probabilities will be later incorporated as parameters in the analytical model to be

developed in this work. We discuss now the influence of the type of code assignment on the errors in the processing of the preamble.

2.2.3 Preamble Code Assignments and Preamble Collisions

Receivers are normally able to lock onto new packets in the presence of the data portion of other packets. Indeed, the data portions are usually encoded using codes different from, and ideally orthogonal to, those of the preambles. Thus, the presence of codes of the data portions will produce, at the output of the filter matched to a given preamble code, a signal that looks like background noise. On the other hand, the presence of the preamble code will give rise to pulses which are narrow and whose amplitude is much higher than that of the background noise, and from which the receiver is able to deduce the bit timing of the desired packet, as well as the information contained in the preamble, and thus lock onto the new packet.

Preamble collisions result from the simultaneous presence, at a receiver in the synchronization stage, of preambles using the same codes. In these circumstances, a receiver will see at the output of the matched filter two or more interleaved sets of correlation spikes. If their time separation is smaller than the width of the main correlation lobe, the synchronization process will fail. Otherwise, if the receiver has enough "intelligence" to recognize this situation and select one of these sets, it will be able to lock onto the corresponding packet. The set of neighboring transmissions that are potential preamble colliders depends on the way preamble codes are assigned to different users. We consider three particular cases. We say that we have *uniform* codes if all packet transmissions in the network use the same preamble codes. In *receiver-directed* codes, each node has an assigned networkwide unique code, which is used by all sources wishing to communicate with that node in the preambles of their transmissions. In *transmitter-assigned* codes, each node

has an assigned networkwide unique code, which it uses in the preambles of all its transmissions.

In the case of uniform code sequences, all the transmissions by all neighbors (i.e., nodes within hearing range) of a given node use the same preamble code sequence, and can thus cause possible collisions. In the case of receiver-directed codes, only those transmissions destined to the node under consideration will use the code sequence that the node is "listening" to. Thus, if the set of preamble codes was chosen to have good correlation properties, only those transmissions will, with high probability, cause preamble collisions. If transmitter-assigned code sequences are used, no two preambles present simultaneously at a receiver will use the same codes, and thus the probability of failure to synchronize to one of these signals is even smaller than in the previous cases. However, this last solution requires the existence, at a node, of a bank of filters matched to the preamble codes of all its neighbors.

2.2.4 Data Portion Code Assignments and Bit Errors

The bits in the data portion of a packet can be encoded using all the same code, or using codes that change on a bit-by-bit basis. We refer to the former situation as *bit-homogeneous* codes, and to the latter as *bit-changing* codes. These two systems possess different behavior with respect to bit errors caused by multiuser interference. Consider a system where bit-homogeneous codes are used. In this case, given a packet locked onto by a receiver, an overlapping packet using the same code will interfere whenever its autocorrelation peaks fall within those of the earlier packet; i.e., whenever the bit periods of the overlapping packets are within a few chip times of each other, causing the peaks to overlap. This means that there is a vulnerable period of a few chip times for each bit during which the arrival

of a new packet will cause destruction. Since usually the number of chips in a codeword is large (from several tens to on the order of a thousand), the cumulative vulnerable period for a packet remains small compared to its entire transmission time, and hence the capture effect. The use of bit-changing codes will lessen the effect of overlapping packets. In this case, the receiver must be equipped with a programmable matched filter which follows the pattern as it varies from bit to bit. If the pattern is long enough so that it does not repeat itself during the transmission of the longest packet, then any late overlapping packet will not interfere with the packet locked onto earlier, since it will not produce any autocorrelation peak during the entire reception time of the earlier packet. The problem of packet acquisition incurred with arrival times too close to each other still persists, since one will require the code sequence to start from the same point, known to receivers.

Different codes can be used in the encoding of the data portions of packets belonging to different immediate source-destination pairs. As in the case of the preamble codes, we distinguish the following cases: *uniform* codes, *receiver-directed* codes, and *transmitter-assigned* codes. The same conclusions can be drawn for each of these assignments regarding the potential for bit errors caused by transmissions with the same codes overlapping the desired packets within the vulnerable period. However, as mentioned above, the extent of this vulnerable period is just a few chip times at the start of a packet in the case of bit-changing codes, and can be neglected for all practical purposes.

The conditions for correct reception of a packet which is locked onto, and during whose reception the destination node does not switch to transmit mode, depend additionally on the forward error correction scheme. We do not discuss this aspect in this work. A study incorporating the effect of coding in the system performance can be found in [Stor85].

2.3 Summary and Conclusions

We examined in this Chapter the two types of signaling used in packet radio, namely narrowband and spread spectrum, and the capture behavior obtained therefrom. Section 2.1 discussed narrowband signaling. The narrowband signal was characterized by the properties that

(i) the presence of a transmission using the modulation scheme in question is easily detected;

(ii) an overlap in time at a receiver with a signal with the same modulation type and of comparable power will give rise to bit errors in the desired data.

A satisfactory approximate model for the capture behavior of narrowband signaling consists of assuming that, if P_D is the received power of the desired signal and P_I is the received power of the interfering signal, a desired data bit will be correctly received if $P_I < \beta P_D$, for some $0 \leq \beta \leq 1$, with β^{-1} being called the *capture ratio*. This behavior is referred to as *power capture*. Even though, due to power capture, individual bits of a packet may be correctly received in the presence of interference, the correct reception of a packet was seen to depend additionally on the ability of the receiver to distinguish the data bits belonging to the interfering packet from those belonging to the desired packet.

Section 2.2 discussed spread spectrum signaling. In this type of signaling, data bits are individually encoded prior to modulating a carrier. Spread spectrum signals were characterized by the properties that

(i) it is difficult for a receiver, without the knowledge of the codes being used in a transmission, to detect their presence;

(ii) spread-spectrum offers a larger degree of immunity than narrowband to

bit errors caused by overlaps with similar signals; these similar signals can be, in particular, time delayed versions of the desired signal, whose presence is due to multiple propagation paths between the transmitter and the receiver (multipath components).

We described the generation of the signal for the *direct sequence* and *frequency-hopping* variants of spread spectrum. We also discussed how the synchronization of a receiver with the codes of the incoming signal can be accomplished by the use of a *preamble*. We then classified the possible choices of code sequences, for use either in the preamble or in the data portion of a packet, according to (i) how sequences are assigned to different users (as *transmitter-assigned*, *receiver-directed*, or *uniform*), and (ii) how codes are assigned to different bits of a given user (as *bit-changing*, or *bit-homogeneous*). We also argued that, in terms of the immunity they afford regarding multiuser interference, bit-changing codes perform better than bit-homogeneous codes, and that, within the latter category, the ranking of types of assignment in terms of descending order of performance is (i) transmitter-assigned codes, (ii) receiver-directed codes, and (iii) uniform codes.

Chapter 3

CHANNEL ACCESS PROTOCOLS

In the previous Chapter we examined one of the aspects of the process of successful exchange of packets between two nodes, namely the conditions under which a receiver will correctly receive the data in a packet in the presence of multiuser interference. In this Chapter we focus on a complementary aspect of that process, which is the conditions under which a transmitter is allowed to transmit a packet. Such conditions define the *channel access protocol.*

In order for a node to make the decision on whether its transmitter can start transmitting a new packet at a given time, the node needs information concerning the state of a subset of the network. For a practical implementation of the protocol, the node has to have the capability to perform a number of functions (such as sensing the activity of neighboring nodes) in order to acquire that information. These functions may or may not be easily achievable, depending in particular on the type of signaling, and on the type of code assignments in the case of spread spectrum signaling. We examine these issues in the present Chapter. Section 3.1 presents a formalization of the notion of channel access protocol. Section 3.2 gives a number of definitions that relate to the specification of the access protocol. Among these will

be the sensing functions just referred to, and for which we will discuss feasibility as a function of the system structure. Section 3.3 classifies the protocols commonly considered for packet radio operation in three classes, according to the nature of the information required for their operation: (i) ALOHA protocols, (ii) activity sensing protocols, and (iii) Busy Tone protocols. It then defines the main protocols of practical interest in each of these classes.

3.1 Introduction

An access protocol is a set of rules that, given the current global state of the network, determines whether or not a given source node can initiate the transmission of a new packet to a given destination node. Formally, let $X_k(t)$ represent a sufficient description, for the purposes of the access protocol, of the state of node k at time t. A complete *network state* $X(t)$ is obtained by building a vector of the node states

$$X(t) = (X_1(t), X_2(t), \ldots, X_N(t)).$$

Let j represent a directed radio link in the network, and L be the number of such links. The access protocol is specified by a vector boolean function

$$\mathbf{B}(X) = (B_1(X), B_2(X), \ldots, B_L(X)),$$

where $B_j(X)$ takes the value 1 if link j is *blocked* (that is, not allowed to initiate a transmission) when the state of the network is X, and 0 otherwise.

The most complete state description of the network contains the history of the network activity up to the current time. However, only a very small subset of this information is usually used by an access protocol. For most protocols of interest,

in an ideal situation of zero propagation delay, a sufficient description $X(t)$ of the network state is one which includes the information, for each node, as to whether the node is currently idle, transmitting a packet, or receiving a packet, and in the last two cases, the destination or source node, respectively. In a more complex situation one might have to include additional information, such as whether the node is receiving the header or the data portion of a packet, whether a busy tone signal is being transmitted, or even some information about the past history of the system. Consider, as an example, the 1-persistent CSMA protocol ([Klein75c]). In this protocol a terminal with a packet ready for transmission senses the channel before attempting transmission. If the channel is sensed idle, the terminal transmits the packet. Otherwise, the terminal waits and then transmits the packet as soon as the channel is sensed idle again. Thus the transmission of a packet is attempted as soon as the channel goes idle whenever there was an attempt at transmitting that same packet during the preceding busy period. The state description for this protocol then requires, in the case of a node whose channel is currently busy, information on whether a packet transmission was attempted during the time interval between the instant the channel last went busy, and the current time. Another example where additional state information is required is a situation of nonzero propagation delay. In such a situation the decision made at node n and time t by the protocol is based on the state of nodes $k = 1, \ldots, N$, at times $t - \tau_{kn}$, where τ_{kn} is the propagation delay from node k to node n. Thus the state description will at least have to contain the past history $\{ X_k(t - \tau_{kn}) : 1 \leq k \leq N, 1 \leq n \leq N \}$. In the model later considered in this work, the state description will have the simpler form described in the first part of this paragraph.

From a conceptual point of view, it is possible to have protocols such that the decisions made at a node make use of global state information. However, for practically realizable protocols, the rules embodied in the access protocol are constrained

to be defined only in terms of information that can be made available locally at the source node of the link, such as the state of the receiver at that node, and the state of the transmitters in some neighborhood of it.

3.2 Definitions

We now give a number of definitions relevant to the specification of the channel access protocol. Some of these definitions relate to the functions through which a node acquires information about the activity of neighboring nodes.

Link: We say that a *link* exists from node k into node n if there is radio connectivity from k to n. Node k is called the *source*, and node n the *destination*, of the link. A link is defined by the ordered pair whose first component is the source node and the second component is the destination node.

Active link: A link is said to be *active* whenever the source node of the link is transmitting a packet to the destination node. (Note that, in general, whenever a packet is transmitted over a link it will be heard at nodes other than the destination node of the link, due to the broadcast nature of the channel, unless directional transmission is employed.)

Active node: A node is defined to be *active* whenever any of the links that emanate from that node is active. Equivalently, the node is active whenever the local transmitter is transmitting a packet.

Locking: Under certain protocols, a receiver at a node may want to transmit a packet even in the presence of activity from the part of neighboring nodes. This behavior makes sense, for example, if there is a good indication that no useful data would be obtained by listening to that activity, either because of interference, or because

no data is intended for the node in question. On the other hand, a node might not want to transmit in a situation where potentially good data can be acquired by listening to the channel. We formalize this idea under the notion of *locking*. We say that a receiver is *locked* onto a packet whenever a packet transmission is present at the receiver, and the receiver does not have a reason to "believe" that it is acquiring useless data. Unfortunately, given the half-duplex nature of the packet radio communication, it is not possible to implement in packet radio systems collision-detection features such as those possible in a broadcast bus, and thus the only circumstances in which a receiver can "refuse" to listen to useless data are (i) in the case of a transmission of which the receiver was not able to correctly receive the initial part (possibly containing the synchronization symbols, source/destination information, etc.), and (ii) in the case that the transmission is not intended to it. It would be desirable to recognize the situation depicted in Figure 2.1a, where a packet that is captured by a receiver is later overlapped by a weaker packet that does not cause bit errors, but whose data continues beyond the end the first packet and can thus be mistaken as belonging to it. A partial capability to detect this situation can be provided if the packet headers contain a "length" field that the receiver can use to detect the end of a packet. After receiving the specified number of characters, a receiver will know that any further data does not belong to the packet being received, and will be able to transit to the unlocked state. Errors in the reception of the header will cause the failure of the above process. However, if the header length is much smaller than the average packet length, the probability of erroneous decision due to incorrect reception of the length field is small.

In a narrowband system, assuming no capture effects, a receiver will have to find an idle channel prior to locking. In a spread-spectrum system the receiver has to synchronize to and receive the preamble of the packet free of errors. This can be done even in the presence of other overlapping transmissions, with a probability

which depends on the correlation properties of the codes employed. In a spread spectrum system there is another mechanism that a receiver can use to determine the end of a desired packet. This mechanism consists of monitoring the codes used. If the interfering signals do not use the same codes (which would be the case with bit-changing codes, or with bit-homogeneous transmitter-assigned codes), or use the same codes with a bit timing separation larger than the width of the "vulnerability window" of the codes used, the receiver can determine the end of the desired packet by observing the decrease in energy at the output of the integrate-and-dump detector circuit.

Code sensing: Code sensing refers to the determination of whether, at a given time, some transmission using a given code sequence is present at a receiver. In the case of bit-homogeneous codes, code sensing can be achieved by the use of filters matched to the desired waveforms. In the case of bit-changing code assignments code sensing is not generally possible, unless one knows exactly the starting time of the desired code sequence.

Activity sensing: Activity sensing refers to the ability of a node to determine whether a given link or group of links is active. In a narrow-band system, *carrier sensing* allows a node to determine whether some neighboring node is active; by *directional* carrier sensing it is possible to determine whether a *given* neighboring node is active. Carrier sensing, by itself, does not allow a node to determine if a given neighboring link (i.e., a link whose source node is a neighbor of the node in question) is active. This goal can be achieved by more sophisticated and expensive means, such as a separate signaling channel, in principle even possessing a connectivity different from that of the data channel. In spread-spectrum systems, the characteristics of activity sensing depend on the particular system characteristics. In the case of systems with uniform code assignment and bit-homogeneous codes,

activity sensing can be done by code sensing, and possesses the same properties as carrier sensing in a narrowband system. In a system with a receiver-directed code assignment and bit-homogeneous codes, activity sensing can be done by simultaneous sensing of the codes of all receivers within a two-hop radius. In this way it is possible to determine whether some neighboring node is active. By directional sensing, it is possible to determine whether a given neighboring node is active, and what the destination is. In transmitter-assigned bit-homogeneous systems, activity sensing can be done by sensing the codes assigned to nodes within a one-hop radius. Furthermore, the identity of the active node(s) can be determined by observing which codes are present. In the case of systems with bit-changing codes, activity sensing is not possible, due to the difficulty of performing code sensing. In such systems, a scheme involving a separate signaling channel can be used to achieve activity sensing.

Busy tone: Some channel access protocols assume the existence of a *busy tone* signal, which nodes send on a separate channel whenever some subset (defined by the particular access protocol considered) of their neighboring links is sensed active. In this way a node can obtain information about the activity of neighbors which are two radio hops away. A busy tone channel can be regarded as a particular case of the separate signaling channel referred to in connection with activity sensing.

3.3 Protocol Classification and Definitions

We now give the definitions of a number of channel access protocols that are commonly considered for use in a radio environment. These protocols can be divided into groups according to different aspects of their operation. One such division concerns whether a node which is locked onto a packet is allowed to abort the

reception in order for a transmission to take place. We call a protocol *disciplined* if this behavior is not allowed to take place, and *undisciplined* otherwise. Another division concerns whether *preemption* is allowed, that is, whether a transmitter can abort the transmission of a packet in order to start the transmission of a new packet. We only consider protocols in which preemption is not allowed. Still another division concerns the nature of the information that is required by the protocol operation. We divide the protocols considered in this work, which are the ones more commonly considered for packet radio applications, into three groups: (i) *ALOHA* protocols, (ii) *activity sensing* protocols, and (iii) *busy tone* protocols.

3.3.1 ALOHA Protocols

The decisions made at a node by the protocols in this family depend only on the state of the node itself. We have basically two protocols in this group:

Pure ALOHA (ALOHA): The source node of a link is allowed to become active whenever it is not transmitting a packet (even if at the time the receiver at the node is locked onto a packet).

Disciplined ALOHA (D-ALOHA): The source node of a link is allowed to transmit whenever it is neither transmitting nor locked onto a packet.

ALOHA was the first protocol considered for packet radio, in the ALOHA packet radio system ([Abra70]). However, since this was a centralized full-duplex system, with separate radio channels for the inbound and outbound traffic, there was no distinction between the disciplined and undisciplined versions of the protocol. This distinction only was deemed necessary for multihop applications, when half-duplex operation of the stations is considered. In these circumstances, the undisciplined version of the protocol has the undesirable feature that it allows the reception of a possibly successful packet to be aborted in order for a scheduled transmission

to take place. The disciplined version corrects this problem. The minimal coordination among the stations in their transmission attempts results in relatively low throughputs for these protocols in a zero-capture fully-connected environment, and this fact motivated the introduction of some of the protocols in the next class.

3.3.2 Activity Sensing Protocols

The decisions made at a node by the protocols in this class depend only on the state of the node, and on the activity of neighboring transmitters (as determined by an activity sensing mechanism).

Carrier Sensing Multiple Access (CSMA): The source node of a link is allowed to transmit whenever neither the node itself nor any neighboring nodes is active.

Destination Code Sensing Multiple Access (DCSMA): This protocol assumes a spread-spectrum environment in which receiver-assigned bit-homogeneous code sequences are used. A transmission over a link is allowed if, at the scheduled time, the source node of the link is neither active nor locked onto a packet, and the presence of the code assigned to the destination node is not sensed.

The last protocol can also be applied in an environment of uniformly assigned bit-homogeneous code sequences, in which case it reduces to CSMA. As mentioned in connection with the feasibility of activity sensing, the protocols of this class do not fit naturally in the framework of a system with bit-changing codes.

CSMA improves dramatically upon the low throughput of ALOHA in a centralized fully-connected environment. Consider, however, a situation where several transmitters, which can not hear each other, can all be heard by the same receiver (*hidden terminals*). CSMA will not help in this case coordinate the transmission attempts of the different transmitters in such a way that at any time only one of them

is trying to access the receiver. The protocols in the next class were introduced as a solution to this problem.

3.3.3 Busy Tone Protocols

The protocols in this family assume the existence of a separate busy tone channel. Whenever a new packet starts transmission, a subset of the nodes that can hear that transmission sends a signal on the busy tone channel in order to insure that only one transmitter at a time tries to access the receiver to which the packet is destined. In the Busy Tone family of protocols, a node is prevented from transmitting whenever it senses activity of neighboring nodes, or it senses a signal on the busy tone channel. Different protocols result from different choices of the set of nodes that set the busy tone.

Conservative Busy Tone Multiple Access (C-BTMA): Whenever a link becomes active, all the neighbors of the source node of the link set the busy tone.

Under C-BTMA, all nodes within one hop of an active transmitter set the busy tone, and all nodes within two hops of it are blocked. In this way, the protocol creates a "buffer zone" around the destination receiver in such a way that, in an environment of zero propagation delay, no two transmissions are ever simultaneously present at the same receiver, thus affording collision-free operation. One can argue, however, that an excessively large number of nodes is blocked by the protocol. Indeed, in the situation of Figure 3.1a, consider that node C transmits, say, to node A. Nodes A, B, D, and F will sense C's activity, set the busy tone, and be blocked. In addition, node E will also sense the busy tone emitted by the other nodes, and be blocked. However, node E, if allowed, could transmit to both B or G without any packet overlaps at a receiver resulting therefrom. Thus one is led to consider

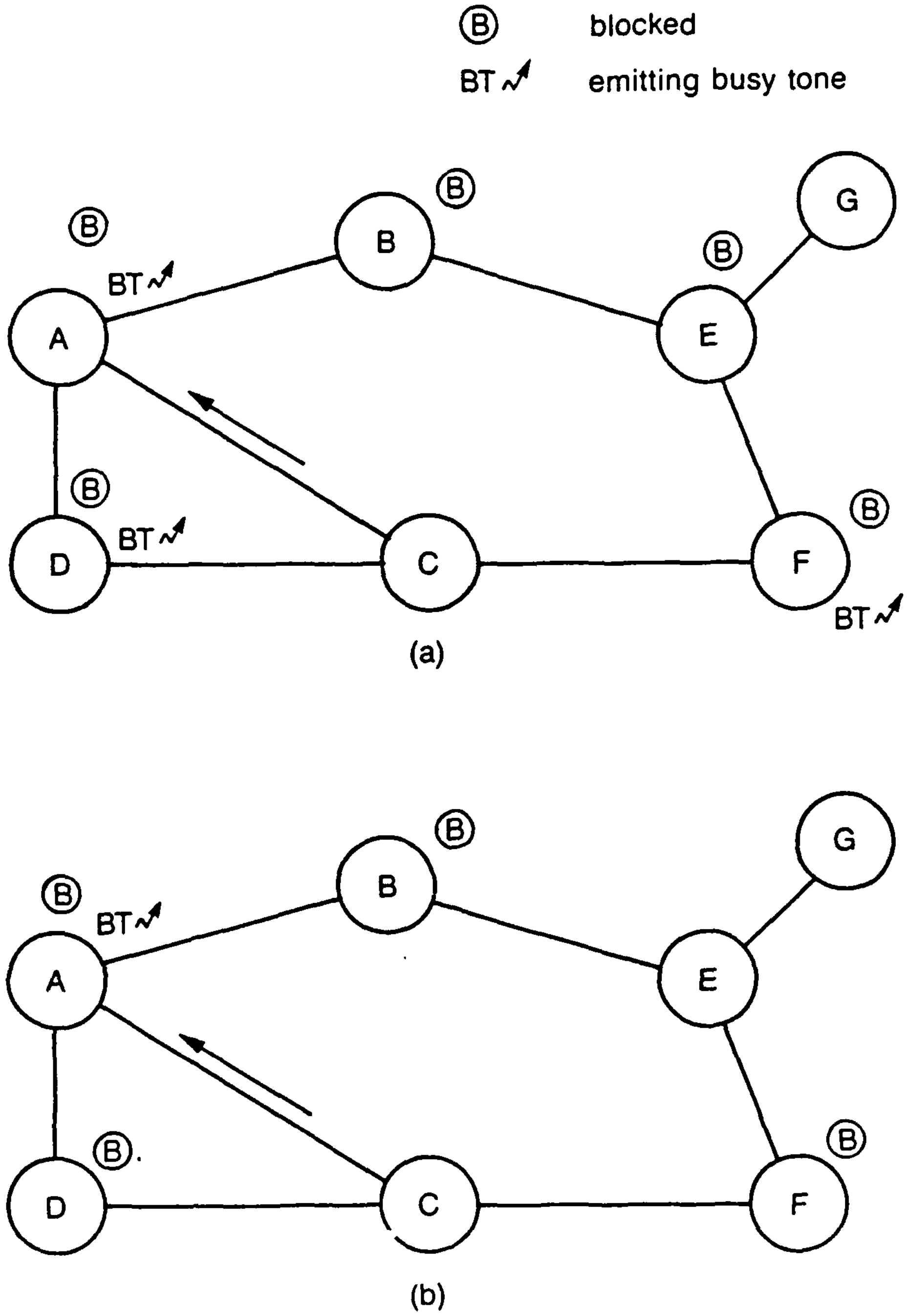


Fig. 3.1 Link blocking in a network with Busy Tone protocols

the situation where only the destination node of the link that becomes active sets the busy tone (Figure 3.1b), as implemented in the following protocols.

Idealistic Destination Busy Tone Multiple Access (ID-BTMA): Whenever a link becomes active, the destination node of the link sets the busy tone.

This protocol assumes that the receiver at a node can determine whether an intended packet is being transmitted to it, even if the packet was not locked onto. This information can be obtained in a bit-homogeneous receiver-directed system by code sensing, but is not otherwise directly available in any of the other forms of signaling discussed (hence the designation *Idealistic*). Another Busy Tone protocol results from a modification of ID-BTMA along these lines. Before giving its definition, let us note that collisions can occur under ID-BTMA. Referring to Figure 3.1b, we see that node E is not blocked after the start of C's transmission to A. If E then transmits to F there will be an overlap of packets at F, since F can hear C's transmission. For signaling schemes such as narrowband the packet will be lost with high probability. However, it is easy to see that ID-BTMA guarantees a reception free of collisions after a packet starts being successfully received. We now introduce a modification of ID-BTMA in which a node only sets the busy tone after locking onto a packet.

Locked Destination Busy Tone Multiple Access (LD-BTMA): Whenever a link becomes active and the destination node successfully locks onto the packet, the destination node sets the busy tone.

Some more elaborate Busy Tone schemes can be envisioned using coded busy tone signals to allow the identification of the originator of the busy tone. A discussion of these schemes, which we shall not consider here, can be found in [Toba85].

3.4 Summary and Conclusions

We discussed in this Chapter channel access protocols. Section 3.1 discussed the kind of information required by a channel access protocol. Section 3.2 gave a number of definitions relevant for the specification of channel access protocols. Among these are the notions of *locking, code sensing,* and *activity sensing*. The feasibility of activity sensing was discussed, and it was concluded that activity sensing is easy to accomplish in narrowband systems or in spread spectrum systems with bit-homogeneous codes, but that it is otherwise difficult for other types of signaling. Section 3.3 classified the protocols commonly considered for packet radio operation in three classes, according to the nature of the information required for the decisions they make at a node: (i) ALOHA protocols, (ii) activity sensing protocols, and (iii) Busy Tone protocols. ALOHA protocols use information concerning only the state of the node itself; activity sensing protocols use information concerning the state of the node in question and of its immediate neighbors; and Busy Tone protocols use information concerning the state of nodes two hops away from the node in question, obtained by means of a separate (busy tone) channel. The implementability of each of these protocols was seen to be subject to the same restrictions, in terms of the signaling method used, as the activity sensing functions required by the protocol.

Chapter 4

THE ANALYTICAL MODEL

We introduce in this Chapter a model for the analytical evaluation of throughput in packet radio systems. Section 4.1 introduces the elements that specify a packet radio network: topology, channel access protocol, and capture mode. It then defines formally the classes of channel access protocols and capture modes considered for analysis. Section 4.2 introduces the network operating model that is considered in the remaining part of this work, and discusses the validity of its assumptions for the purpose of capacity analysis.

4.1 Model Elements

4.1.1 Topology and Traffic Requirements

We consider a packet radio network with N nodes, numbered $1, 2, \ldots, N$, which utilize a single radio broadcast channel. The topology of the network is given by a hearing matrix $\mathbf{H} = [h_{ij}]$, where

$$h_{ij} = \begin{cases} 1 & \text{if } j \text{ can hear } i \\ 0 & \text{otherwise.} \end{cases}$$

Thus each nonzero entry h_{ij} in the hearing matrix corresponds to a directed radio link in the network from node i to node j, and vice-versa. We call node i the *source* and node j the *destination* for that particular link. This "hard" model for the hearing is a simplification of the situation found in a real system, in which there exists a continuum of grades between full hearing and no hearing. However, this representation does not detract from the generality of the model, since we shall incorporate later parameters that represent the quality (in terms of the probability of bit error) of the network links.

The traffic requirements for each link are assumed to be dictated by the end-to-end traffic requirements together with a static routing function. It may happen that for some links the required traffic is zero. We refer to these links as *unused links,* and to all other links as *used links.* As in Section 3.2, we say that a used link is *active* whenever a transmission is taking place over that link, i.e., when the source node is transmitting a message intended to the destination node on that link. We say that an active link is *locked onto* whenever the destination node of the link is locked onto the packet that is being transmitted. We consider all used links to be numbered $1, 2, \ldots, L$, and we let $\mathcal{L} \triangleq \{1, 2, \ldots, L\}$. For link $i \in \mathcal{L}$, we denote by $s(i)$ its source node, and by $d(i)$ its destination node. Alternatively we represent link i by the ordered pair $(s(i), d(i))$.

4.1.2 Channel Access Protocol

Section 3.1 presented a definition of a channel access protocol that is too general to be useful for analysis. We shall restrict our attention to a class of access protocols that will allow analytical tractability. This class is still general enough, however, to include most protocols of practical interest in packet radio applications, in particular those defined in Section 3.3.

The class of access protocols considered in this thesis is defined by the following two conditions:

1. **State description:** The decisions made by the protocol use only information concerning which links are active and which links are locked onto.

This condition means that a sufficient description of the state of a given node k consists of the information on whether the node is idle, transmitting a packet (and over which link), or locked onto a packet (and over which link the packet is being received). A convenient way of representing this information is by having the state $X_k(t)$ of node k at time t be defined by

$$X_k(t) = \begin{cases} 0, & \text{if } k \text{ is idle,} \\ +i, & \text{if link } i \in E(k) \text{ is active,} \\ -j, & \text{if link } j \in V(k) \text{ is locked onto,} \end{cases} \tag{4.1}$$

where $E(k)$ is the set of links that has k as source node (i.e., the set of emanating links), and $V(k)$ is the set of links that has k as a destination node (i.e., the set of incoming links).

2. **Structure of blocking:** For each link i there exist two sets of links $\mathcal{B}_a(i)$ and $\mathcal{B}_l(i)$, such that link i is blocked if and only if some link in $\mathcal{B}_a(i)$ is active, or some link in $\mathcal{B}_l(i)$ is active and locked onto. Formally, this condition is equivalent to requiring that the boolean function $B_i(\cdot)$ which describes the blocking of link i (see Section 3.1) have the form

$$B_i(X_1, X_2, \ldots, X_N) = \bigvee_{k=1}^{N} \Big(X_k \in \mathcal{B}_a(i) \vee (-X_k) \in \mathcal{B}_l(i) \Big),$$

where X_k is defined by Equation (4.1).

In condition 2., the sets $\mathcal{B}_a(i)$ and $\mathcal{B}_l(i)$ are specified by the channel access protocol. Since there are several possible choices for $\mathcal{B}_l(i)$ leading to the same access protocol,

we further adopt the convention that $\mathcal{B}_l(i)$ is such that $\mathcal{B}_l(i) \cap \mathcal{B}_a(i) = \phi$. In other words, $\mathcal{B}_l(i)$ is formed by those links that block i whenever they are locked onto, but do not block i when they are active and not locked onto.

A subclass of the class of protocols defined by conditions 1. and 2. above possesses special analytical properties, and will be studied separately. These protocols lead to a form of decoupling between the description of the activity of the transmitters and the activity of the receivers. We designate this class by $\mathcal{D}$, and refer to the corresponding systems as *decoupled systems.* This class is defined by the following condition:

Protocol class $\mathcal{D}$: A protocol is of class $\mathcal{D}$ if the decisions made by the protocol use only information concerning which links are active (or, equivalently, if $\mathcal{B}_l(i) = \phi$). Formally, this condition is equivalent to having

$$B_i(X_1, X_2, \ldots, X_N) = \bigvee_{k=1}^{N} (X_k \in \mathcal{B}_a(i)) .$$

As an example, ID-BTMA is in class $\mathcal{D}$. Indeed, in ID-BTMA the knowledge that link i is active allows one to infer which nodes or links are blocked: all nodes that can hear i are blocked, and all nodes that can hear the busy tone sent by the destination node of link i are also blocked. Other protocols in $\mathcal{D}$ are ALOHA, CSMA, and C-BTMA. LD-BTMA is an example of a protocol not in $\mathcal{D}$. In LD-BTMA, whenever link i is active, all nodes that can hear i's source are blocked, but the nodes that can hear i's destination are blocked only if this node set the busy tone, that is, only if the destination node locked onto i's transmission. Another protocol not in $\mathcal{D}$ is D-ALOHA.

For protocols of class $\mathcal{D}$, we say that link $i \in \mathcal{L}$ blocks link j if, whenever link i is active, the protocol used does not allow a scheduling point of link j to result

in an actual transmission. It is to be noted that if link i blocks link j, it does not necessarily follow that link j blocks link i. An example of this situation is provided by ID-BTMA in a 4-node chain, as described in Example 5.1.7.

Let D be a set of links in $\mathcal{L}$. We say that D blocks link $j \in \mathcal{L} - D$ if there exists some link $i \in D$ which blocks j.

4.1.3 Capture Mode

In this study, we consider the operation of the receivers to be described by the following model:

1. The length of the preambles is small enough, compared with the length of the packets, to be neglected.

2. The receiver at node n, when idle (i.e., not locked onto a packet), will listen for the preambles of packets transmitted by any of the links in the network, and will successfully lock onto the preamble of a packet from incoming link j with probability $\mathcal{P}_n(D; j)$, independently from trial to trial. Here D is the set of links which are active just before the transmission of link j's packet starts, and $\mathcal{P}_n(D; j)$ equals the average probability of success for the system with preambles of nonzero length, given D and j, and averaged over all possible evolutions of the system during the preamble's transmission time. As long as the length of the preamble is much smaller than the average rescheduling interval and the average packet length one can consider $\mathcal{P}_n(D; j)$, as an approximation, to be independent of these parameters.

3. If the preamble is successfully received, the receiver remains locked onto that packet until the end of its transmission, except if, in the meanwhile, the node to which the receiver belongs switched to transmit mode.

4. While the receiver at node n is locked onto link j's packet, bit errors occur at the output of the receiver at a rate $\epsilon_n(D;j)$, where D the set of active links.

5. If the preamble is *not* successfully received, the receiver will remain idle, searching for a new packet to lock onto, and the whole process is repeated from step 2.

6. A packet is successfully received if it is successfully locked onto and the data portion is received free of errors.

By appropriately specifying the families of parameters $\mathcal{P}_n(D;j)$ and $\epsilon_n(D;j)$ we can model a wide variety of capture situations. As special cases of capture that can be represented in this way we have the following two cases.

Zero capture: Under zero capture, an idle receiver is only able to lock onto a new packet if no neighboring nodes are active. After locking onto a packet, the receiver will receive the packet successfully if and only if, during its reception, no neighboring nodes of the receiver ever become active.

Idealistic perfect capture: Under idealistic perfect capture, an idle receiver is only able to lock onto a new packet if no neighboring nodes are active. After locking onto the packet, the receiver will receive the packet successfully, independently of the activity of the neighboring nodes.

Zero capture had already been discussed in Chapter 2 in connection with narrowband systems. Idealistic perfect capture corresponds to the limit case of a system with uniformly assigned bit-homogeneous codes, with the same preamble and data portion codes, as the number of chips per bit goes to infinity.

A subclass of the class of capture modes defined by conditions 1.–6. above will be seen to lead to a simplification in the throughput computations. This class, designated by $\mathcal{C}_s$, is defined by the following condition:

Capture class $\mathcal{C}_s$: A capture mode belongs to class $\mathcal{C}_s$ if, for any link j, the

availability of the destination receiver to lock onto a packet transmitted on that link is completely determined by the set D of links that are active just before the transmission of the packet starts.

An example of a capture mode that belongs to class C_s is zero capture. Under zero capture, the knowledge that a neighboring node of the destination of link j is active at the time link j starts transmission of a new packet assures that the packet will not be locked onto, either because the receiver is locked onto a previous packet and thus not available for the new one, or the receiver is idle, but also unable to lock onto a new packet given the definition of zero capture. Conversely, the knowledge that all neighboring nodes of a given destination node, and the node itself, are idle, obviously implies that the destination receiver is free to lock onto a new packet. Similarly, it is easy to see that idealistic perfect capture also belongs to class C_s. As an example of a capture mode not belonging to class C_s, consider the case where receivers can lock onto new packets even in the presence of neighboring activity. Then the set of active nodes by itself does not give information on whether or not the receiver is available to lock onto a new packet, since additional information is required about whether or not the receiver is already locked onto some other packet at the time the new packet starts.

4.2 Network Operating Assumptions

Since the entire packet radio network operates using a single radio frequency, each node in the network has one transmitter, but can in general have more than one outgoing link. We consider that each outgoing link at a node has a separate queue for the packets to be transmitted on it and that the transmitter is shared among all queues at that node. It is assumed in this study that neither preemption

nor priority functions are supported at the nodes. To avoid repeated interference between transmissions in the network, transmission requests for the various queues at a node are scheduled according to random point processes, one for each queue.

Consider a point in time defined by the point process for some link i. If the queue is empty, this scheduling point is ignored. If the queue is nonempty then a packet in the queue is considered for transmission. The transmission may or may not take place depending on the status of the source node (namely, idle, transmitting or receiving), the channel access protocol in use, and the current activity in the network. If the transmission is inhibited, or if the transmission is undertaken but unsuccessfully (due to a collision at the intended destination), then the packet in question is reconsidered at the next scheduling point in time. Otherwise (i.e., the transmission is successful), the packet is removed from the queue, and the same process is repeated at the next scheduling point for that link.

We present now the system assumptions considered in the analytical model. These are identical to the assumptions introduced by Boorstyn and Kershenbaum in [Boor80].

1. **Infinite queues:** The queues at each outgoing link have infinite storage capacity.
2. **Heavy traffic:** At each scheduling point of the scheduling point process there is a packet in the queue for consideration.
3. **Acknowledgments:** Instantaneous and perfect acknowledgments.
4. **Delays:** Zero propagation and processing delays.
5. **Scheduling processes:** The rescheduling point process for link i, $i \in \mathcal{L}$, is Poisson with rate λ_i ($\lambda_i > 0$), and independent of all other such processes in the network.
6. **Packet length distribution:** The transmission time of the messages transmitted over link i is exponentially distributed with mean $1/\mu_i$ ($\mu_i > 0$).

7. **Redrawing of packet lengths:** The transmission time of the messages transmitted over link i is redrawn independently from the corresponding distribution each time the message is transmitted.

The above assumptions discard, for reasons of tractability and as discussed in Section 1.2, some of the aspects of the operation of a packet radio system. For the purposes of capacity analysis, however, the resulting model represents the essential aspects, in particular the two main goals (i) action of the channel access protocol, and (ii) effect of the capture mode. Furthermore, assumptions 1.–7. do not put excessively artificial requirements on the system. Assumptions 1. and 2. fit naturally in a capacity analysis framework. By introducing assumption 3., we obtain upper bounds on the performance of the system with nonideal acknowledgments. More importantly, it should be possible to infer the ordering of the protocols based on their relative performance in the system with nonideal acknowledgments from the results obtained for the ideal case. The first part of assumption 4. discards the effects resulting from making local decisions based on delayed information. However, those will be second order effects as long as the propagation delays are much shorter than the average times that the system spends in a given state, which will be the case if the propagation delays are much smaller than the average packet durations and the average rescheduling delays. In a typical packet radio system, with ranges of the order of 20 km, bits of duration $10\mu s$ each, and packet lengths of the order of 1000 bits, the ratio of the propagation delay to the packet duration is of the order of 7×10^{-3}, which we neglect. Assumption 5. results from the previous assumptions, in particular 2., 3., and 4., by considering a scheduling algorithm which at given times looks at the packet at the head of the queue, and schedules it for transmission after a random exponentially distributed scheduling delay with parameter λ_i for link i. The times at which the algorithm looks at the packet at the head of the queue are (i) at the end of a packet transmission, or (ii) at the time of an already

existing scheduling point. In case (i), if the packet was successful (which is known instantaneously, according to 3.) then the new packet at the end of the queue is scheduled. Otherwise the same packet is rescheduled. In case (ii), a transmission of the packet at the head of the queue is attempted. The transmission will be undertaken if so allowed by the channel access protocol. Otherwise the packet is rescheduled for later transmission. From this process, clearly a Poisson rescheduling process results. Assumption 6. reflects a particular choice for the distribution of the length of the packets on the channel which will allow tractability. It will be relaxed later in Chapter 7. Assumption 7. deserves some remarks. In systems where the probability of success of a message depends upon its length (such as a zero capture system), larger packets require on the average a larger number of transmissions than do shorter packets. As a consequence, the random variable representing the length of the packets present on the channel is stochastically larger than the random variable representing the length of the successful packets; that is, the graph of the distribution function of the former lies on or below the graph of the distribution function of the latter. This effect had already been reported by Ferguson in [Ferg77] where, in a single-hop system, it was studied what the length distribution of the successful packets should be in order for a given length distribution of the channel packets to result. In our case, as will be seen later, it is possible, although computationally time-consuming, to take a similar approach, by determining what the value of the μ_i's should be in order for the successful packets to have the desired average length. It will not be possible, in general, to enforce an a *priori* given length distribution. We shall not, however, perform those computations. If the ratios of the average lengths of the successful packets to those of the channel packets do not differ very much over the set of the network links, the results derived should be a good approximation to those obtained by "tuning" the μ_i's to yield the desired average successful packet length. The model derived from

the assumptions presented in this Section seems thus appropriate for the purpose of capacity analysis.

4.3 Summary and Conclusions

We presented in this Chapter the analytical model to be considered in the remaining part of this work. Section 4.1 defined formally the elements of the model: (i) topology and traffic requirements, (ii) channel access protocol, and (iii) capture mode. Section 4.1.1 described the topology and traffic requirements by means of a hearing matrix and a traffic matrix, respectively. Section 4.1.2 defined the class of access protocols to be considered for analysis: all those protocols whose decisions make use only of information concerning which links are active and which links are locked onto. This class of protocols includes the Pure ALOHA, Disciplined ALOHA, CSMA, C-BTMA, ID-BTMA, and LD-BTMA protocols. This section further defined a subclass (designated as *class* $\mathcal{D}$) of the above class of channel access protocols, for which the protocol decisions are based only on the state of the network transmitters, and whose elements lead to a form of decoupling in the representation of the network activity and in the evaluation of the network throughput. Examples of protocols in this class are Pure ALOHA, CSMA, C-BTMA, and ID-BTMA. Subsection 4.1.3 described the general capture model considered for analysis. Under this model, a capture mode is described by (i) the (state-dependent) probability of an idle receiver locking onto a packet destined to it, and (ii) the (state-dependent) rates at which bit errors occur at a receiver during the reception of a packet. Section 4.3 introduced the network operating model that is considered in the remaining part of this work. The essential features of this model are: (i) a situation of heavy traffic, where packets are always available for consideration at a scheduling point,

(ii) Poisson rescheduling processes, (iii) exponential packet lengths, (iv) redrawing of message lengths at each retransmission attempt, (v) instantaneous and perfect acknowledgments, and (vi) zero propagation and processing delays. In this Section we also discussed the validity of the model assumptions for the purpose of capacity analysis, and concluded for their appropriateness.

Chapter 5

DECOUPLED SYSTEMS

We present in this Chapter the analysis of the class of protocols for which the protocol decisions are completely specified by the state of the network transmitters. The protocols in this class lead to a form of decoupling between the description of the activity of the receivers and the activity of the transmitters, and the corresponding systems are referred to as *decoupled systems*. Examples of such protocols are Pure ALOHA, CSMA, C-BTMA, and ID-BTMA. Section 5.1 presents a Markovian model for the description of the transmitter activity. This model is a direct extension of a model introduced by Boorstyn and Kershenbaum in [Boor80] for the analysis of Carrier Sense Multiple Access. In this Section we characterize the state space of the stochastic process describing the transmitter activity, derive the balance equations for its steady-state probability distribution, and investigate the conditions under which this distribution possesses a product form solution. The existence of a product form solution is shown to be equivalent to symmetry in link blocking. In this Section we also show that the product form solution is computationally NP-hard. Section 5.2 presents a Markovian model for the description of the activity of the receivers, and derives the equilibrium equations of the corresponding stochastic process in a

form that emphasizes the decoupling afforded by the class of protocols considered in this Chapter. Section 5.3 presents the derivation of throughput measures from probability measures and hitting times associated with the processes describing the activity of the receivers and of the transmitters. This Section also presents the derivation of the throughput equations for a four-node ring network, as an example of application of the formalism developed in this Chapter.

5.1 Markovian Description of Transmitter Activity

In this Section we give the analysis of the transmitter activity under protocols of class $\mathcal{D}$. For this class of protocols, the set of active links contains all the information needed for the protocol decisions. Thus, *for the purpose of the description of the protocol activity*, we can take the state $X(t)$ of the network to be the set of active links.

Let the state of the system at time t be $D \in \mathcal{S}$, let i be any link not blocked by D, and let $j \in D$. Given the assumptions in Section 4.2, the time until the next scheduling point of i is exponentially distributed with parameter λ_i, and the time to the end of the transmission over link j is also exponentially distributed with parameter μ_j. Given that $X(t) = D$, and given the memoryless property of the exponential distribution, the state of the system at time $t + \Delta t$ is completely determined, in a probabilistic sense, by the state of the system at time t, and so $\{X(t)\}$ is a continuous time Markov chain. In this next Section we study the properties of this Markov chain.

5.1.1 State Space

We now define the state space $\mathcal{S}$ for the Markov chain $\{X(t)\}$. Since $X(t)$ is the set of all links that are active at time t, $\mathcal{S} \subseteq 2^{\mathcal{L}}$. Given an access protocol and its blocking properties, not all subsets of $\mathcal{L}$ may be in $\mathcal{S}$.

Definition 5.1.1 $\mathcal{S}$ is the collection of subsets of $\mathcal{L}$ that the system can reach starting from the idle state ϕ (i.e., all links inactive) by any sequence of *link activations* and *deactivations.*

Definition 5.1.2 A subset $D = \{l_1, l_2, \ldots, l_n\}$ of $\mathcal{L}$ is said to be *directly reachable* if there exists some permutation $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ of D such that l_{i_j} is not blocked by $(l_{i_1}, l_{i_2}, \ldots, l_{i_{j-1}})$, $j = 2, \ldots, n$. That is, D is directly reachable if it can be reached by *only* activating the links in it, in some order, starting from the idle state ϕ.

Lemma 5.1.3 *If a subset $D = \{l_1, l_2, \ldots, l_n\}$ is directly reachable, then any subset $D' \subseteq D$ is also directly reachable.*

Proof: Let $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ be an ordered sequence of activations which allows D to be reached. The ordered subsequence in $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ corresponding to links in D' is a sequence of activations which allows D' to be reached directly. ∎

Proposition 5.1.4 *The state space $\mathcal{S}$ consists of ϕ and all subsets $D \subseteq \mathcal{L}$ that are directly reachable.*

Proof: Clearly a set D which is directly reachable belongs to $\mathcal{S}$. To prove the converse, we let $D \in \mathcal{S}$ be some subset that is reached via some sequence of states $D_0, D_1, \ldots, D_m$, with $D_0 = \phi$ and $D_m = D$, due to link activations and deactivations. (Note that since the process $\{X(t) : t \geq 0\}$ is such that no two events can occur at the same instant, then $|D_k| = |D_{k-1}| \pm 1$ for all $k = 1, 2, \ldots, m$). Since the first transition out of $D_0 = \phi$ must be an activation, there is some index $r \leq m$ such

that D_r is reached directly. Consider D_{r+1}. If $D_{r+1} = D_r \cup \{i\}$ for some i, then D_{r+1} is clearly directly reachable. If $D_{r+1} = D_r - \{j\}$ for some j, then D_{r+1} is also directly reachable, by Lemma 5.1.3. Applying the same argument to the remaining steps, we guarantee that D is directly reachable. ∎

According to Proposition 5.1.4, one can generate the state space with the algorithm below. This algorithm can however be very inefficient. More efficient algorithms shall be presented in Chapter 8.

begin

  $\mathcal{S} := \{\phi\}$;

  $\mathcal{L} := \{1, 2, \ldots, L\}$;

  for $k := 0$ **to** $L - 1$ **do**

    for every $D \in \mathcal{S}$ **such that** $|D| = k$ **do**

      for every $l \in \mathcal{L} - D$ **do**

        if l **is not blocked by** D**, then add** $D \cup \{l\}$ **to** $\mathcal{S}$**;**

end.

Throughout the analysis we assume a fixed ordering of the state space $\mathcal{S}$, according to which the rows and columns of all the vectors and matrices to be considered are indexed.

Remark 5.1.5 Given an access protocol and some state $D \in \mathcal{S}$, it should be noted that not all sequences of activations of its elements will necessarily allow D to be reached from ϕ. For example, consider the 4-node chain of Figure 5.1 with nonzero traffic requirement over links 1 and 5 only, and the ID-BTMA access protocol. State $\{1, 5\}$ is an example of a state for which the order of activation is relevant. This state is reachable by the permutation $(1, 5)$, but not by the permutation $(5, 1)$.

Remark 5.1.6 Recall that $\mathcal{L}$ is the set of all used links and thus $\lambda_i > 0$ for all $i \in \mathcal{L}$. Accordingly every state can be reached from the empty state in a nonzero

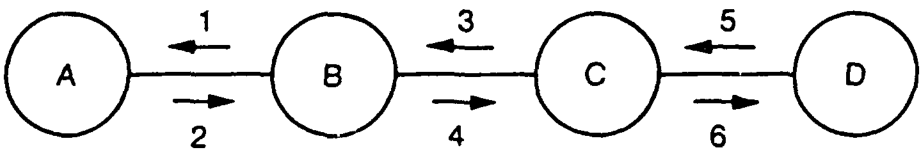


Fig. 5.1 A 4-node chain

period of time with nonzero probability. Similarly, the empty state can be reached from any other state in a nonzero period of time with nonzero probability (since $\mu_i > 0$ for all $i \in \mathcal{L}$). It then follows that all states communicate and the resulting Markov chain is irreducible.

5.1.2 The Equilibrium Equations

As noted above, the Markov chain $\{X(t) : t \geq 0\}$ is irreducible. Since the state space is finite, the chain is then positive recurrent and ergodic. Thus the existence and uniqueness of a stationary distribution is ensured. We denote by $\{p(D) : D \in \mathcal{S}\}$ the stationary probability distribution, and let $\mathbf{p} = \Big(p(D)\Big)_{D \in \mathcal{S}}$ be the row vector of the steady-state probabilities.

Let D be a set of links in $\mathcal{L}$. We define $U(D)$ to be the set of all links in $\mathcal{L} - D$ which are not blocked by D. Let the state of the system at time t be $D \in \mathcal{S}$, and let $i \in U(D)$ and $j \in D$. Given the assumptions in Section 4.2, the time to the next scheduling point of i is exponentially distributed with parameter λ_i, and the time to the end of the transmission over link j is also exponentially distributed with parameter μ_j. Thus, given that $X(t) = D$, link $i \in U(D)$ can become active at a rate λ_i, and link $j \in D$ can become inactive at a rate μ_j. The state of the system

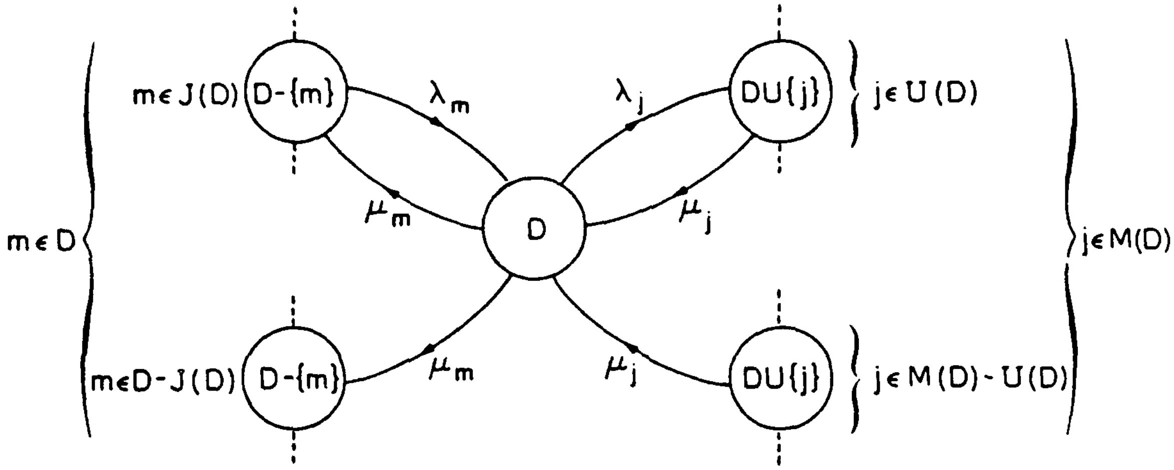


Fig. 5.2 Typical transitions to and from a state

at time $t + \Delta t$ is then given by

$$X(t+\Delta t) = \begin{cases} D \cup \{i\}, & i \in U(D), \text{with probability } \lambda_i \Delta t + o(\Delta t) \\ D - \{j\}, & j \in D, \text{with probability } \mu_j \Delta t + o(\Delta t) \\ D, & \text{with probability } 1 - \Big(\sum\limits_{i \in U(D)} \lambda_i + \sum\limits_{j \in D} \mu_j \Big) \Delta t + o(\Delta t) \end{cases}$$

This equation defines the transition rates which we need for writing the equilibrium equations. Before doing so we have to introduce some further notation. For each $D \in \mathcal{S}$, let $M(D)$ be the set of all links $i \notin D$ such that $D \cup \{i\} \in \mathcal{S}$. Clearly $M(D) \supseteq U(D)$. Note however that it is not necessarily true that $M(D) = U(D)$. (See Example 5.1.7 below.) Let $J(D)$ to be the set of all links $j \in D$ such that j is not blocked by $D - \{j\}$, i.e., such that $j \in U(D - \{j\})$. Clearly, $J(D) \subseteq D$. Here too, in general we have $J(D) \neq D$, as is also illustrated in Example 5.1.7. With these definitions, a sketch of the state-transition-rate diagram for state D and the transitions to and from its neighbors can be seen in Figure 5.2. An equivalent

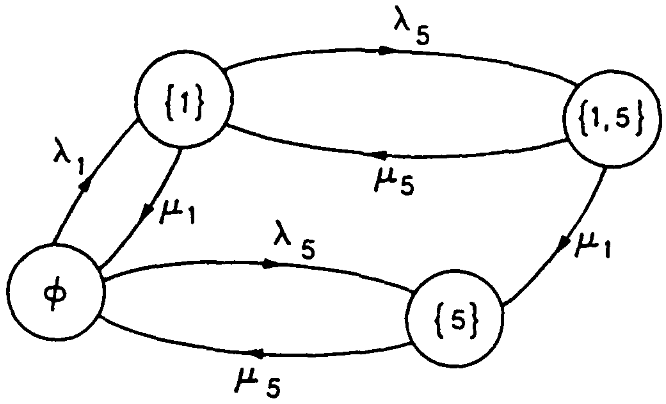


Fig. 5.3 State Space for Markov chain of Example 5.1.7

description is given by the transition-rate matrix $\mathbf{Q} = [q(D, D')]_{D,D' \in S}$, where

$$q(D, D') = \begin{cases} \lambda_i, & \text{if } D' = D \cup \{i\},\ i \in U(D) \\ \mu_j, & \text{if } D' = D - \{j\},\ j \in D \\ -\left(\sum\limits_{i \in U(D)} \lambda_i + \sum\limits_{j \in D} \mu_j \right), & \text{if } D' = D \\ 0, & \text{otherwise} \end{cases}$$

The equilibrium equations take then the form

$$p(D) \left[\sum_{i \in U(D)} \lambda_i + \sum_{j \in D} \mu_j \right] = \sum_{j \in J(D)} p(D - \{j\})\, \lambda_j + \sum_{i \in M(D)} p(D \cup \{i\})\, \mu_i, \quad D \in S \tag{5.1}$$

Example 5.1.7 Consider the 4-node chain of Figure 5.1 with nonzero traffic requirement over links 1 and 5 only, and the ID-BTMA protocol. The corresponding state-transition-rate diagram is shown in Figure 5.3. From the definitions we have that $J(\{1,5\}) = \{5\}$, $U(\{5\}) = \phi$, and $M(\{5\}) = \{1\}$. These are examples of states D for which $M(D) \neq U(D)$, or $J(D) \neq D$.

5.1.3 Reversible Markov Chains and Product Form Solutions ([Kell79])

Definition 5.1.8 A continuous time stochastic process $\{X(t)\}$ defined in $\mathcal{I} = (-\infty, +\infty)$ is said to be *reversible* if for any $\tau \in \mathcal{I}$, integer n, and $t_1 \leq t_2 \leq \cdots \leq t_n$ in $\mathcal{I}$, $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$.

For the particular case of Markov chains, reversibility has a simple characterization in terms of the transition rates and steady-state distribution, as given in the following proposition, whose proof can be found in [Kell79].

Proposition 5.1.9 *A stationary continuous time Markov chain is reversible if and only if there exists a collection of positive numbers $\{\gamma(D) : D \in \mathcal{S}\}$, summing to unity, such that*

$$\gamma(D_1) \cdot q(D_1, D_2) = \gamma(D_2) \cdot q(D_2, D_1) \tag{5.2}$$

for all $D_1, D_2 \in \mathcal{S}$, and where $q(D_i, D_j)$ is the rate of transitions from D_i to D_j. When such a collection exists, it is the stationary probability distribution.

An equivalent necessary and sufficient condition for reversibility (called Kolmogorov's criterion) is that, for any finite sequence of states $D_1, D_2, \ldots, D_n \in \mathcal{S}$, the transition rates satisfy

$$q(D_1, D_2)\, q(D_2, D_3) \cdots q(D_n, D_1) = q(D_1, D_n)\, q(D_n, D_{n-1}) \cdots q(D_2, D_1). \tag{5.3}$$

Suppose we are given a reversible Markov chain with state space $\mathcal{S}$. Let D_0 be a fixed state and D a generic state in $\mathcal{S}$. Let $D_0, D_1, \ldots, D_m$ be any sequence of states in $\mathcal{S}$, with $D_m = D$, such that between any two consecutive states of the sequence there exist nonzero transition rates. By repeated application of (5.2) it is

easy to see that the steady-state probability distribution for such a Markov chain satisfies

$$p(D) = p(D_0) \prod_{k=1}^{m} \frac{q(D_{k-1}, D_k)}{q(D_k, D_{k-1})}. \tag{5.4}$$

A solution with the form of (5.4) is called a *product form* solution. It is immediately seen that if the steady-state solution satisfies (5.4), then (5.2) is automatically satisfied for all $D_1, D_2 \in \mathcal{S}$. Thus

Proposition 5.1.10 *A stationary continuous time Markov chain $\{X(t) : t \geq 0\}$ possesses a product form solution for the steady-state probability distribution if and only if it is reversible.*

5.1.4 Criterion for the Existence of a Product Form

We use here the results of the previous Section to determine the conditions on the access protocol, network topology, and traffic requirements under which the resulting Markov chain is reversible, and hence the global balance equations (5.1) have a product form solution.

Lemma 5.1.11 *$U(D) = M(D)$ for all $D \in \mathcal{S}$ if and only if $J(D) = D$ for all $D \in \mathcal{S}$.*

Proof: We know already that $J(D) \subseteq D$ and $U(D) \subseteq M(D)$. To prove the desired equalities we only need to prove the reverse inclusions. Assume that $U(D') = M(D')$ for all $D' \in \mathcal{S}$. It is evident that $J(\phi) = \phi$. Consider now any $D \in \mathcal{S}, D \neq \phi$. For each $j \in D$, by definition $j \in M(D - \{j\})$. Since by hypothesis $U(D - \{j\}) = M(D - \{j\})$, then $j \in U(D - \{j\})$. But this just means that $j \in J(D)$. Thus $D \subseteq J(D)$, for all $D \in \mathcal{S}$. Conversely, assume that $D' = J(D')$ for all $D' \in \mathcal{S}$. Call a state *maximal* if $M(D) = \phi$. Since $U(D) \subseteq M(D)$, for maximal states it is true that $U(D) = M(D)$. Let now $D \in \mathcal{S}$ be a non-maximal state, and $j \in M(D)$. By hypothesis $J(D \cup \{j\}) = D \cup \{j\}$, which in particular implies that j is not blocked by D, and thus that $j \in U(D)$. Hence $M(D) \subseteq U(D)$. ∎

Proposition 5.1.12 *The Markov chain $\{X(t) : t \geq 0\}$ is reversible if and only if*

$$D = J(D) \tag{5.5}$$

for all $D \in \mathcal{S}$ (or, equivalently, $U(D) = M(D)$ for all $D \in \mathcal{S}$).

Proof: Assume that the Markov chain is reversible. Clearly (5.5) holds for $D = \phi$. Consider now $D \in \mathcal{S}, D \neq \phi$, and $j \in D$. From (5.2) we have that

$$p(D) \cdot q(D, D - \{j\}) = p(D - \{j\}) \cdot q(D - \{j\}, D).$$

Since $q(D, D - \{j\}) = \mu_j > 0$ and $p(D) > 0$ for all $D \in \mathcal{S}$, this last equation implies that $q(D - \{j\}, D) > 0$. But since $q(D - \{j\}, D)$ can only be either 0 (if $j \notin J(D)$) or λ_j (if $j \in J(D)$), we necessarily conclude that $q(D - \{j\}, D) = \lambda_j$ and $j \in J(D)$. Then $D \subseteq J(D)$ for all $D \in \mathcal{S}$, and consequently $D = J(D)$ for all $D \in \mathcal{S}$. Conversely, assume that $J(D) = D$ for all $D \in \mathcal{S}$. We now show that $\{\gamma(D) : \gamma(D) = \gamma_0 \prod_{i \in D} \frac{\lambda_i}{\mu_i}, D \in \mathcal{S}\}$, with γ_0 chosen so that $\sum_{D \in \mathcal{S}} \gamma(D) = 1$, is a collection of numbers that satisfies the conditions of Proposition 5.1.9. Let D_1, D_2 be any two states in $\mathcal{S}$. Assume first that they are of either the form $D_1 = D$, $D_2 = D - \{j\}$, or the form $D_1 = D - \{j\}$, $D_2 = D$, for some $D \in \mathcal{S}$ and $j \in D$. From the choice of $\gamma(D)$ we have

$$\gamma(D) = \frac{\lambda_j}{\mu_j} \gamma(D - \{j\}).$$

The transition rates between these two states are $q(D, D - \{j\}) = \mu_j$ and, from the assumption $J(D) = D$, $q(D - \{j\}, D) = \lambda_j$. Thus, in this case,

$$\gamma(D_1)q(D_1, D_2) = \gamma(D_2)q(D_2, D_1).$$

For any other choice of D_1 and D_2, $q(D_1, D_2) = q(D_2, D_1) = 0$, and

$$\gamma(D_1)q(D_1, D_2) = \gamma(D_2)q(D_2, D_1)$$

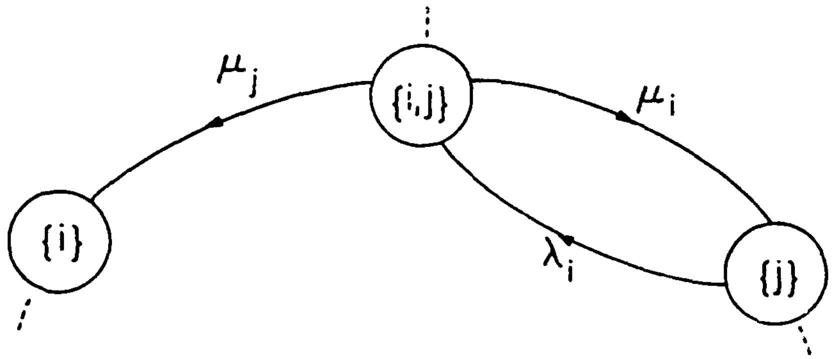


Fig. 5.4 Portion of a nonreversible chain

is trivially verified. Thus (5.2) holds for all $D_1, D_2 \in \mathcal{S}$, and $\{X(t) : t \geq 0\}$ is reversible, by Proposition 5.1.9. ∎

Proposition 5.1.13 (Criterion for the existence of a product form—exponential packet length distributions) *A necessary and sufficient condition for a channel access protocol, together with a given network topology and traffic requirements, in a system with exponential packet lengths, to have a product form solution is that, for all pairs of used links i and j, link j blocks link i whenever link i blocks link j.*

Proof: We will prove the equivalence between the condition stated in the above criterion and the condition that $J(D) = D$, for all $D \in \mathcal{S}$.

(a) $J(D) = D$, for all $D \in \mathcal{S}$.

Assume that link i blocks link j. If j does not block i, we will have the situation depicted in Figure 5.4 in which $j \in \{i,j\}$ but $j \notin J(\{i,j\})$, providing an instance of a state D for which $J(D) \neq D$, which is a contradiction. Thus j blocks i.

(b) There exists D such that $J(D) \neq D$.

Since $J(D) \subseteq D$ and $J(D) \neq D$, there exists $j \in D$ such that j is blocked by $D - \{j\}$. Let $i \in D - \{j\}$ be some link blocking j and define $D' = \{i,j\}$. Since $D' \subseteq D$ then, by Lemma 5.1.3 and Proposition 5.1.4, D' belongs to $\mathcal{S}$, and D' can

be directly reached by activating links i and j in some order. By hypothesis i blocks j, and so D' has to be directly reachable from $\{j\}$. Thus j does not block i. ∎

Proposition 5.1.13 implies that, in a reversible chain and for any state $D \in \mathcal{S}$, *any* order of activation of the links in D allows D to be reached directly from state ϕ, and thus the situation depicted in Remark 5.1.5 does never occur.

For a reversible chain the stationary probability distribution is given by (5.4). From the particular form of the transition rates we have

$$p(D) = p(\phi) \prod_{i \in D} \frac{\lambda_i}{\mu_i} \tag{5.6}$$

for all $D \in \mathcal{S}$. We can ask if there can exist protocols for which the corresponding Markov chain $\{X(t) : t \geq 0\}$ is not reversible, and yet the steady-state probabilities have the form (5.6).

Proposition 5.1.14 *(5.6) is a solution of the global balance equations (5.1) if and only if*

$$D = J(D) \tag{5.5}$$

for all $D \in \mathcal{S}$ (or, equivalently, $U(D) = M(D)$ for all $D \in \mathcal{S}$).

Proof: Assume that $D = J(D)$ for all $D \in \mathcal{S}$. By Proposition 5.1.12, $\{X(t) : t \geq 0\}$ is reversible and thus the steady-state probabilities have the form (5.6). Conversely, assume that (5.6) is a solution of (5.1). By substitution of (5.6) in (5.1) and simplification we obtain

$$\sum_{i \in M(D) - U(D)} \lambda_i = \sum_{j \in D - J(D)} \mu_j \, .$$

We now seek the conditions under which this equality can hold. Recall that a state D of the Markov chain is said to be *maximal* if $M(D) = \phi$. Given a generic state D,

define a *maximal path* starting at D to be a finite sequence of states $D_0, D_1, \ldots, D_k$ such that $D_0 = D$, $D_{l+1} = D_l \cup \{i\}$ for some $i \in M(D_l)$, $l = 0, 1, \ldots, k-1$, and D_k is a maximal state. Define the length of the maximal path to be k, and let $l(D)$ be the maximum of the lengths of the maximal paths starting at D. We shall now prove (5.5) by induction on $l(D)$. For $l(D) = 0$ we have that D is a maximal state, for which $M(D) = U(D) = \phi$. Then

$$\sum_{j \in D - J(D)} \mu_j = 0 .$$

Since, by assumption, $\mu_j > 0$, we obtain that $D = J(D)$. Assume now that, for n a positive integer, (5.5) holds for all states D' for which $l(D') \leq n$. Let D be a state for which $l(D) = n + 1$. For all $j \in M(D)$, $D \cup \{j\}$ is a state for which $l(D) \leq n$. By the induction hypothesis we then have $J(D \cup \{j\}) = D \cup \{j\}$, which means in particular that j is not blocked by D or, in other words, that $j \in U(D)$. Then $U(D) = M(D)$ and

$$\sum_{j \in D - J(D)} \mu_j = 0 .$$

Again, as all $\mu_j > 0$, it follows that $D = J(D)$. ∎

Example 5.1.15 As an application of Proposition 5.1.13, we can now prove that, with a symmetric hearing matrix, nonpersistent CSMA always leads to a product form solution. Consider any two used links i and j, and represent them as $(s(i), d(i))$ and $(s(j), d(j))$, respectively. Under CSMA, if i blocks j, then either $s(i) = s(j)$ or $h_{s(i)s(j)} = 1$. The symmetry of the hearing matrix then implies that j blocks i, and thus by Proposition 5.1.13 the stationary distribution will have a product form. If the hearing matrix is not symmetric we will not get a product form solution, except when all pairs of nodes $s(i)$ and $s(j)$ for which $h_{s(i)s(j)} = 1$ and $h_{s(j)s(i)} = 0$ are such that at least one element of the pair is the source of no used links.

Example 5.1.16 The ID-BTMA protocol will not, in general, lead to a product form solution. Indeed, if the network under consideration contains the subnetwork and traffic pattern of Example 5.1.7, we can find links i and j such that i blocks j but j does not block i. For some specific topologies and traffic patterns, however, ID-BTMA will have a product form solution. Examples of these are a star network with arms of length 1 and arbitrary traffic pattern, or a 4-node chain in which the outer nodes generate no traffic.

5.1.5 Computational Complexity

It has been the unwritten experience of the researchers in the field that, even in the case where an analytical solution such as the product form of Equation (5.6) exists for the steady-state probabilities, its determination is computationally hard. We study in this Section the computational complexity of the normalizing factor $p(\phi)^{-1}$ needed to evaluate (5.6).

From the condition $\sum_{D \in S} p(D) = 1$ it follows that the normalizing factor is given by

$$p(\phi)^{-1} = \sum_{D \in \varphi} \prod_{i \in D} \frac{\lambda_i}{\mu_i}. \tag{5.7}$$

A straightforward method of computing the normalizing factor consists of enumerating the state space S and computing $p(\phi)^{-1}$ via Equation (5.7). Proposition 5.1.13 asserts that, when a product form solution exists, the blocking between links is symmetric, and this in turn implies that the state space S is formed only by those states D such that, for each such D, no two links $i, j \in D$ block each other. Thus the problem of the enumeration of the state space is equivalent to the problem of finding all independent subsets of the *link blocking graph* of the network, that is,

the undirected graph with adjacency matrix $A = [a_{ij}]$, where

$$a_{ij} = \begin{cases} 1, & \text{if } i \text{ blocks } j, \text{ and } i, j \text{ are used links,} \\ 0, & \text{otherwise,} \end{cases}$$

problem that is known to be NP-complete. As pointed out in [Magl83], this straightforward method of computing $p(\phi)^{-1}$ is computationally hard.

We show in this Section that any method of computing $p(\phi)^{-1}$ is computationally hard, by showing that the INDEPENDENT SET problem of computational complexity theory ([Gare79]) reduces in polynomial time to the problem of computing $p(\phi)^{-1}$, and thus that the latter problem is at least as hard as the former, known to be NP-complete.

The INDEPENDENT SET problem is defined in terms of an undirected graph $G = (V, E)$, where V is the set of vertices (or nodes), and E is the set of edges (or arcs) of the graph, and is formulated as a decision problem, in a form consistent with the computation model of a Turing machine. We shall also define the problem of computing the normalizing factor as a decision problem on a graph. The graph we are interested in is the link blocking graph of the network in question.

Given a graph G, we say that a set of nodes is an *independent set* if no two nodes in the set are adjacent. The two problems of interest for us are defined as follows.

P1 (INDEPENDENT SET) Given an undirected graph $G = (V, E)$, and an integer $K \leq |V|$, where $|V|$ denotes the cardinality of V, determine whether there is an independent set of vertices $V' \subseteq V$, with $|V'| \geq K$.

P2 (SP) Given an undirected graph $G = (V, E)$, a collection $\{\lambda_v\}_{v \in V}$ of integers, and an integer R, determine whether or not

$$SP(G) \triangleq \sum_{\substack{D \subseteq V \\ D \text{ indep.}}} \prod_{i \in D} \lambda_i \geq R.$$

The input strings to these problems use the symbols "0", "1", and ",". Let $L = |V|$. The input to problem (P1) is $r_1, r_2, \ldots, r_L, K$, where r_j is the j-th row of the adjacency matrix of G. The length of this input is $n_1 = (L+1)\,L + \lceil \log K \rceil$ (all logarithms being taken to be to the base 2), satisfying $L^2 < n_1 < L^2 + 2\,L$. The input to problem (P2) is $r_1, \ldots, r_L, \lambda_1, \ldots, \lambda_L, R$, of length $n_2 = L\,(L+1) + \sum_{i=1}^{L}(\lceil \log \lambda_i \rceil + 1) + \log R$. Note that, since $SP(G) \le \prod_{i=1}^{L}(1+\lambda_i)$, the problem of determining the value of $SP(G)$ can be solved by doing a binary search on $\{1, \ldots, \prod_{i=1}^{L}(1+\lambda_i)\}$, taking at most $\sum_{i=1}^{L}\log(1+\lambda_i) < n_2$ calls to a solver for problem (P2).

We now have

Proposition 5.1.17 *The* **INDEPENDENT SET** *problem reduces in polynomial time to problem* **SP**.

Proof: Given an instance of (P1), create an instance of (P2) with the same adjacency matrix, and with $\lambda_1 = \ldots = \lambda_L = \lambda = 2^L + 1$ and $R = \lambda^K$. The length of the corresponding input string is

$$n_3 = L\,(L+1) + L\,(K+2) + K\,(L+1) \le L\,(L+1) + L\,(L+2) + L\,(L+1) < 8\,L^2,$$

so that $n_3 < 8\,n_1$.

Suppose the largest independent set has at most $K-1$ elements. Then

$$SP(G) = \sum_{\substack{D \subseteq V \\ D \text{ indep.}}} \lambda^{|D|} = \sum_{j=0}^{K-1} \sum_{\substack{D \in V \\ D \text{ indep.} \\ |D|=j}} \lambda^j \le \sum_{j=0}^{K-1} \binom{L}{j} \lambda^j$$
$$< \sum_{j=0}^{K-1} \binom{L}{\lfloor L/2 \rfloor} \lambda^j \approx \sqrt{\frac{2}{\pi L}}\, 2^L\, \frac{\lambda^K - 1}{\lambda - 1} < \lambda^K = R.$$

The two last steps are justified, respectively, by Sterling's approximation, and by the fact that $\lambda \ge 2^L + 1$ implies $2^L\,(\lambda^K - 1)/(\lambda - 1) < \lambda^K$. Supposing now that

there exists an independent subset with K elements, then $SP(G) > \lambda^K = R$. Thus we found a set of parameters, $\lambda_1, \ldots, \lambda_L, R$, function of K and L, such that the answer to an instance of (P2) is "yes" if and only if the answer to the corresponding instance of (P1) is "yes," and the translation from (P1) to (P2) can be done in polynomial time. ∎

5.2 Markovian Description of Receiver Activity

We present in this section the analysis of the activity of the network receivers under protocols of class $\mathcal{D}$. The results derived will later be used for the analysis of the network throughput.

5.2.1 Markovian Representation of Spread Spectrum Systems

For the protocols in class $\mathcal{D}$, the state of a receiver at a given time t is determined by the state of the receiver at time $t = 0$ and by the activity of the transmitters up to time t. It then follows that the addition to the state description $X(t)$ of Section 5.1 of the information regarding the state of time t of the receiver at node n is sufficient to describe the activity of this node. There is thus decoupling between the description of the activity of different receivers, in the sense that the description of the activity at one receiver does not require the knowledge of the activity at different receivers, as opposed to the case of more general protocols, where the description of link activity requires that the state of *all* receivers in the network be recorded in the state description of the system.

Let $r_n(t)$ represent the number of the link that the receiver at node n is locked onto at time t ($r_n(t)$ will be defined as equal to 0 if the receiver is not locked onto

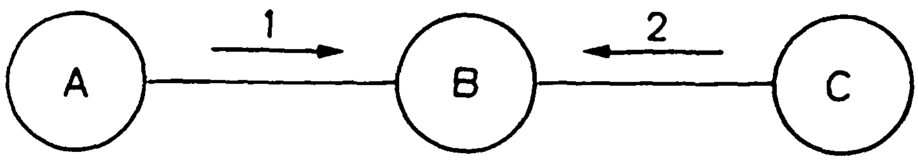


Fig. 5.5 A three-node network

any packet at time t). $\{r_n(t) : t \geq 0\}$ is a continuous time stochastic process. In order to avoid notational ambiguity, we denote the state vector associated with node n by $Y_n(t)$, and define it to be

$$Y_n(t) \triangleq (X(t); r_n(t)).$$

There exists one such process associated with each node in the network. It is clear that, with assumptions 1.–5. of Section 4.1.3., for any $s < t$, $r_n(t)$ is a (random) function of only $r_n(s)$ and $\{X(u) : s \leq u \leq t\}$. Thus, given the Markovian nature of $X(t)$, $Y_n(t)$ is also Markovian, for each n.

In order to define the state space S_n^* of $Y_n(t)$, let $E(n)$ be the set of links *emanating* from n, i.e., the set of links whose source node is n, and define

$$\mathcal{E}(n) = \{D \in S : D \cap E(n) \neq \phi\}.$$

Whenever $X(t) \in \mathcal{E}(n)$, the transmitter at node n will be active, and thus $r_n(t) = 0$. The only states in S_n^* with first coordinate $D \in \mathcal{E}(n)$ are thus the states $(D; 0)$. Let $V(n)$ be defined as the set of the incoming links into node n, that is, the set of links that have n as destination node. If $X(t) \notin \mathcal{E}(n)$, $r_n(t)$ will in general take additional values in $D \cap V(n)$, although some such states $(D; j)$ might not be reachable from state $(\phi; 0)$. An example of this situation is provided by the network of Figure 5.5, where only links 1 and 2 are used, where it is assumed that

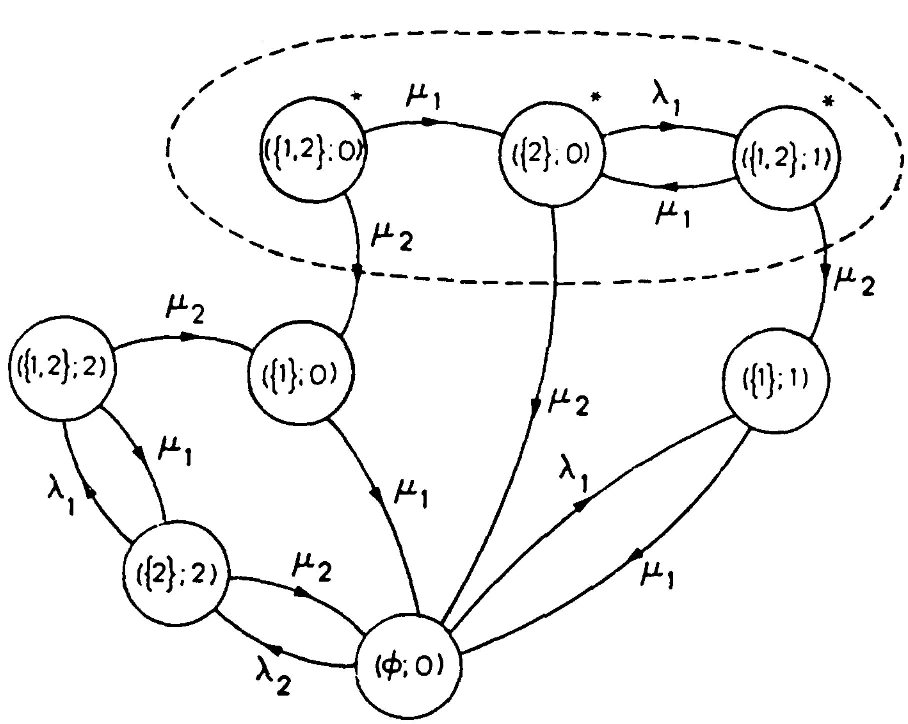


Fig. 5.6 State space for the example of Section 5.2.1

$\mathcal{P}_B(\phi;1) = \mathcal{P}_B(\phi;2) = \mathcal{P}_B(\{1\};2) = \mathcal{P}_B(\{2\};1) = 1$, and where the protocol is such that link 1 blocks link 2, but link 2 does not block link 1. The state transition rate diagram of the corresponding $Y_B(t)$ process is shown in Figure 5.6. The starred states are clearly transient states, not reachable from $(\phi;0)$, and thus will have zero probability.

We define the state space $\mathcal{S}_n^*$ of $Y_n(t)$ to be

$$\mathcal{S}_n^* = \{(D;0) : D \in \mathcal{E}(n)\} \cup \{(D;j) : D \in \mathcal{S} - \mathcal{E}(n)\,, j \in \{0\} \cup (D \cap V(n))\}\,.$$

5.2.2 Existence of Steady-State Probabilities

Although not every two states in $\mathcal{S}_n^*$ communicate, we can still guarantee the existence of a unique steady state distribution. To see that this is indeed the case,

let us first observe that the state $(\phi; 0)$ can be reached from any other state with nonzero probability in any finite time $t > 0$. Let us define $\mathcal{R}_n$ to be the collection of all states which can be reached from $(\phi; 0)$, and $\mathcal{T}_n$ to be the collection of states which cannot be reached from $(\phi; 0)$. $\mathcal{R}_n$ is an ergodic class, since it is formed by a closed class of communicating states, with a finite number of elements. $\mathcal{T}_n$ is a transient class, since there is a *nonzero* probability of reaching state $(\phi; 0)$ from any of its states, and hence never returning to it. It then follows ([Heym82], Corollary 7–9, Th. 7–10) that the Markov chain $Y_n(t)$ possesses a unique steady-state distribution. We let $\{p_n(D; j) : (D; j) \in \mathcal{S}_n^*\}$ denote this distribution.

5.2.3 Balance Equations and Steady State Probabilities

The general form of the balance equations for $Y_n(t)$ is

$$p_n(D; i) \sum_{(D'; i') \in \mathcal{S}_n^*} q_n((D; i), (D'; i')) = \sum_{(D'; i') \in \mathcal{S}_n^*} p_n(D'; i') q_n((D'; i'), (D; i)),$$

where $q_n((D; i), (D'; i'))$ represents the rate of transitions of $Y_n(t)$ from state $(D : i) \in \mathcal{S}_n^*$ into state $(D'; i') \in \mathcal{S}_n^*$. Defining, for a fixed $D \in \mathcal{S}$,

$$\mathcal{I}_D = \{j : (D; j) \in \mathcal{S}_n^*\}$$

to be the set of values that the second coordinate of $(D; \cdot) \in \mathcal{S}_n^*$ can take, the balance equations take the form

$$p_n(D; i) \sum_{D' \in \mathcal{S}} \sum_{i' \in \mathcal{I}_{D'}} q_n((D; i), (D'; i')) = \sum_{D' \in \mathcal{S}} \sum_{i' \in \mathcal{I}_{D'}} p_n(D'; i') q_n((D'; i'), (D; i)). \tag{5.8}$$

For the class of protocols that we are considering we have that, for $D, D' \in \mathcal{S}$ and $i \in \mathcal{I}_D$,

$$P\{X(t + \Delta t) = D' \mid X(t) = D, r_n(t) = i\} = P\{X(t + \Delta t) = D' \mid X(t) = D\}.$$

On the other hand,

$$P\{X(t+\Delta t) = D' \mid X(t) = D, r_n(t) = i\}$$
$$= \sum_{i' \in \mathcal{I}_{D'}} P\{X(t+\Delta t) = D', r_n(t+\Delta t) = i' \mid X(t) = D, r_n(t) = i\}$$

so that, by equating the right hand sides of the previous two Equations, dividing by Δt and taking the limit as $\Delta t \to 0$, we obtain (cf. also [Keme66], Section 6.3)

$$\sum_{i' \in \mathcal{I}_{D'}} q_n((D;i),(D';i')) = q(D,D'), \quad i \in \mathcal{I}_D. \tag{5.9}$$

Using (5.9) in (5.8) and adding the equations (5.8) for all i corresponding to a fixed $D \in \mathcal{S}$ yields

$$\left[\sum_{i \in \mathcal{I}_D} p_n(D;i)\right] \sum_{D' \in \mathcal{S}} q(D,D') = \sum_{D' \in \mathcal{S}} \sum_{i' \in \mathcal{I}_{D'}} p_n(D';i') \sum_{i \in \mathcal{I}_D} q_n((D';i'),(D;i))$$

or

$$\left[\sum_{i \in \mathcal{I}_D} p_n(D;i)\right] \sum_{D' \in \mathcal{S}} q(D,D') = \sum_{D' \in \mathcal{S}} \left[\sum_{i' \in \mathcal{I}_{D'}} p_n(D';i')\right] q(D',D). \tag{5.10}$$

By the law of total probability we have that

$$P\{X(t) = D\} = \sum_{i \in \mathcal{I}_D} P\{Y_n(t) = (D;i)\}.$$

Taking the limit as $t \to \infty$, and due to the uniqueness of the steady state distributions of $X(t)$ and $Y_n(t)$ we obtain

$$p(D) = \sum_{i \in \mathcal{I}_D} p_n(D;i). \tag{5.11}$$

Thus (5.10) becomes

$$p(D)\sum_{D'\in\mathcal{S}} q(D,D') = \sum_{D'\in\mathcal{S}} p(D')\,q(D',D) \tag{5.12}$$

which is the balance equation for state D in the Markov chain $X(t)$. We thus can, for a fixed $D \in \mathcal{S}$, group together the balance equations (with respect to $Y(t)$) for the states $(D;i)$, $i \in \mathcal{I}_D$, and replace an arbitrary one (say, the one corresponding to state $(D;i_0)$), with the balance equation (5.12) for state D (with respect to $X(t)$). Equation (5.11) can then be used to express the probability of the state corresponding to the chosen equation in terms of $p(D)$ and $\{Q_n(D;i), i \in \mathcal{I}_D, i \neq i_0\}$. This procedure allows us to separate the computation of the steady-state probability distribution of $X(t)$, which relates to the activity of the transmitters, from the computation of the steady-state probabilities concerning the activity of the receivers in the network which is another form of the decoupling referred to previously.

We now examine the transition rates of $Y_n(t)$, and write the corresponding equations. As described by (5.9), the sum of the transition rates out of state $(D;i)$ depends only on D. These transitions are illustrated in Figure 5.7 for states $D \notin \mathcal{E}(n)$. The diagram for the transitions out of state $(D;0)$, $D \in \mathcal{E}(n)$, is similar to Figure 5.7a, the differences being that state $(D \cup \{i\};i)$ does not exist, and that thus $\mathcal{P}_n(D;i) = 0$. The transitions into state $(D;j)$ depend on whether the set D considered is in $\mathcal{E}(n)$ or not.

(i) For $D \in \mathcal{E}(n)$, the only corresponding state in $\mathcal{S}_n^*$ is $(D;0)$, the transitions into which are shown in Figure 5.8. The balance equation derived from this diagram and Figure 5.7 will be, upon performing the procedure described in the previous section, replaced by the balance equation for state D (with respect to $X(t)$), and will not be explicitely written here.

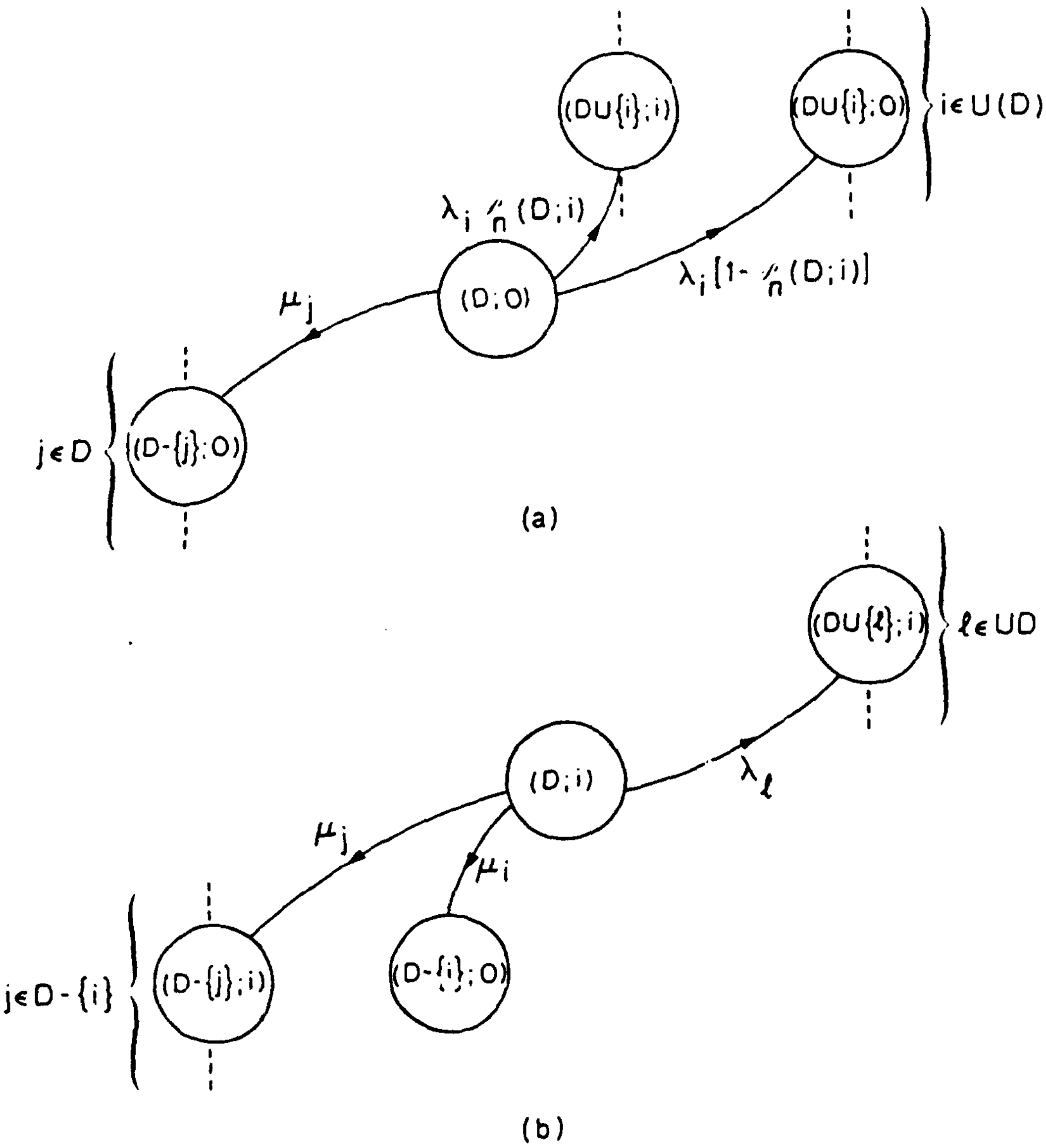


Fig. 5.7 Transitions out of state $(D,\cdot)$, $D \notin \mathcal{E}(n)$.

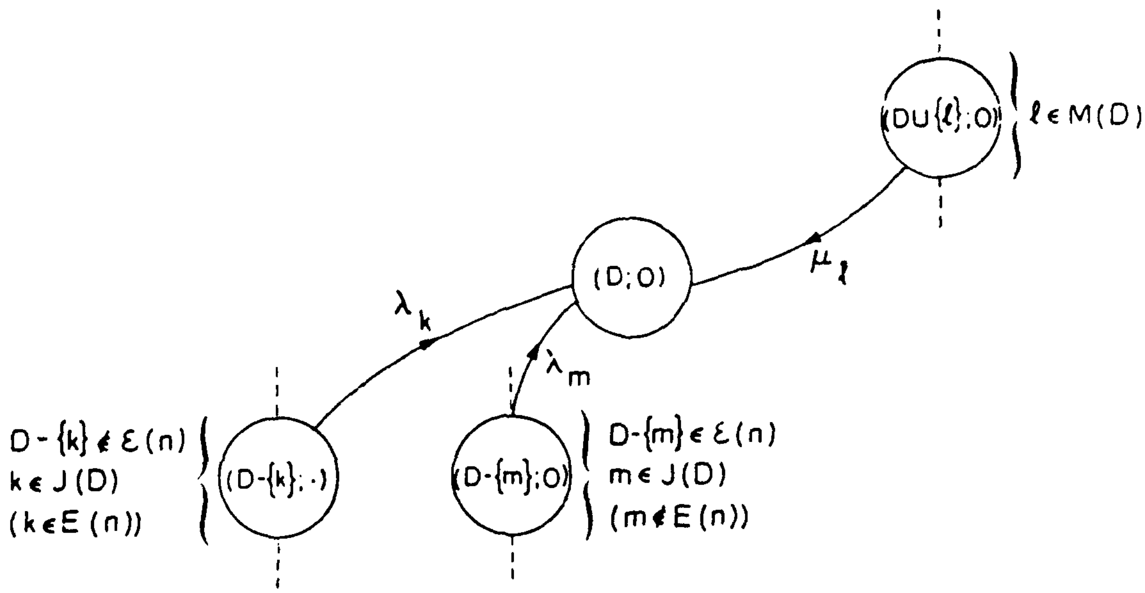
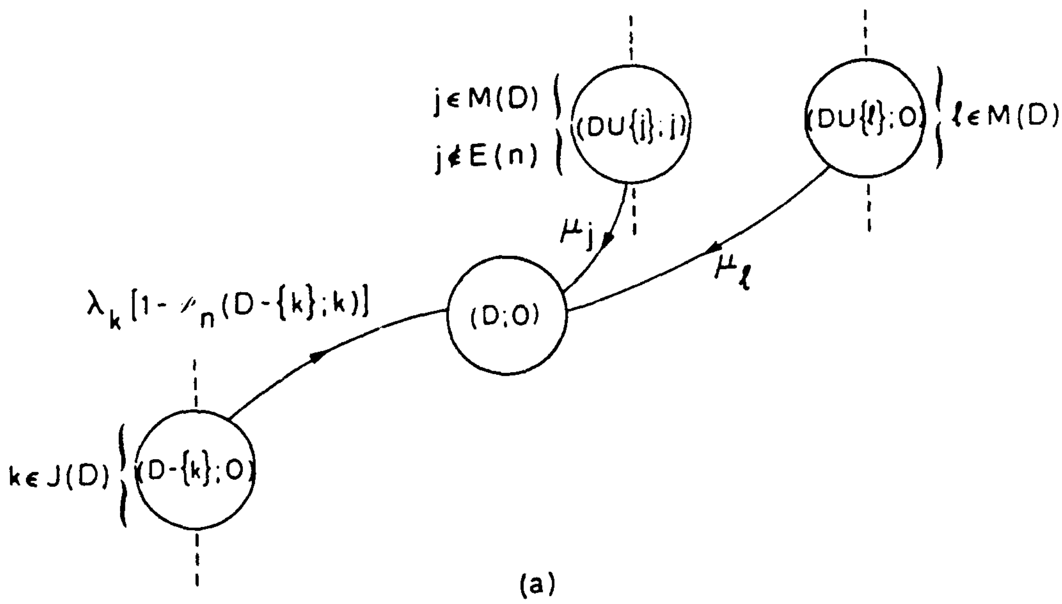
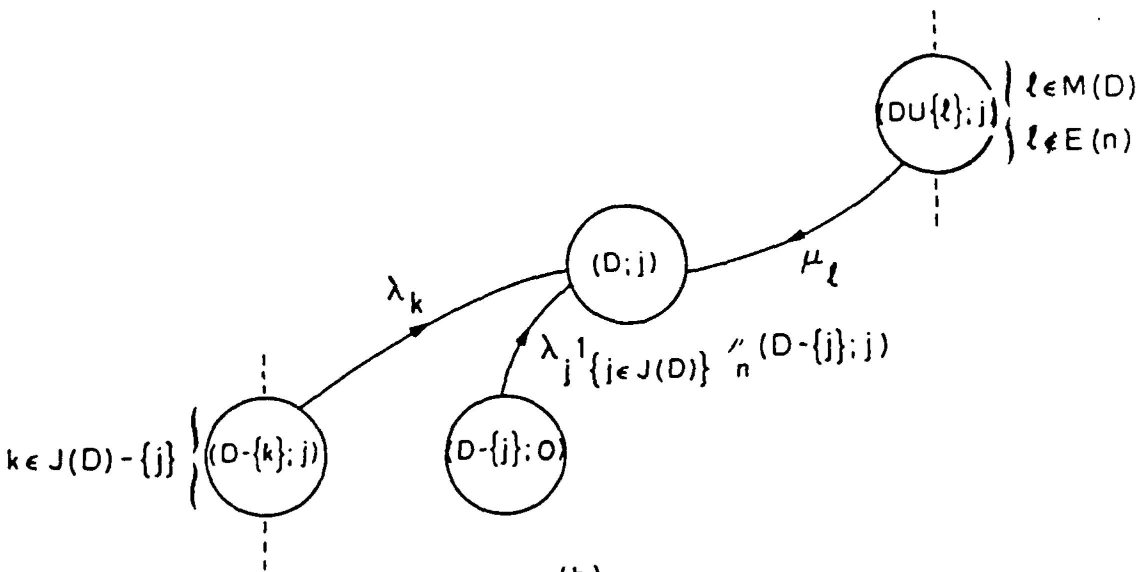


Fig. 5.8 Transitions into state $(D, 0)$, $D \in \mathcal{E}(n)$

(ii) For $D \notin \mathcal{E}(n)$ the cases $(D; 0)$ and $(D; j)$, $j \in \mathcal{I}_{\mathcal{D}} - \{0\}$ have to be considered separately. Figure 5.9a shows the transitions into state $(D; 0)$. The balance equation derived from this diagram and Figure 5.7 will be replaced by the balance equation for state D (with respect to $X(t)$), and will not be written here. The transitions into state $(D; j)$, where $D \in \mathcal{S}$ and $j \in V(n) \cap D$, are shown in Figure 5.9b. The corresponding balance equation is, after eliminating $p_n(D - \{j\}; 0)$ by the use of eq. (5.11),



(a)



(b)

Fig. 5.9 Transitions into state $(D,\cdot)$, $D \notin \mathcal{E}(n)$

$$p_n(D;j)\left[\sum_{k\in D}\mu_k+\sum_{l\in U(D)}\lambda_l\right]$$

$$=1_{\{j\in J(D)\}}\left[p(D-\{j\})-\sum_{\substack{m\in D-\{j\}\\ m\in V(n)}}p_n(D-\{j\};m)\right]\lambda_j\,\mathcal{P}_n(D-\{j\};j)$$

$$+\sum_{k\in J(D)-\{j\}}p_n(D-\{k\};j)\,\lambda_k$$

$$+\sum_{\substack{l\in M(D)\\ l\notin E(n)}}p_n(D\cup\{l\};j)\,\mu_l,\qquad D\in\mathcal{S},\quad j\in D\cap V(n).$$

(5.13)

Introducing the conditional probabilities $\{P_n(D;i),\, D\in\mathcal{S},\, i\in\{0\}\cup(D\cap V(n))\}$, defined by

$$P_n(D;i)\triangleq\frac{p_n(D;i)}{p(D)},\tag{5.14}$$

equation (5.13) can be written as

$$p(D)\left[\sum_{k\in D}\mu_k+\sum_{l\in U(D)}\lambda_l\right]P_n(D;j)$$

$$=1_{\{j\in J(D)\}}\,p(D-\{j\})\left[1-\sum_{\substack{m\in D-\{j\}\\ m\in V(n)}}P_n(D-\{j\};m)\right]\lambda_j\,\mathcal{P}_n(D-\{j\};j)$$

$$+\sum_{k\in J(D)-\{j\}}p(D-\{k\})\,P_n(D-\{k\};j)\,\lambda_k$$

$$+\sum_{\substack{l\in M(D)\\ l\notin E(n)}}p(D\cup\{l\})\,P_n(D\cup\{l\};j)\,\mu_l,\qquad D\in\mathcal{S},\quad j\in D\cap V(n).$$

(5.15)

Once (5.1) has been solved for $\{p(D) : D\in\mathcal{S}\}$, the solutions to the systems of equations (5.13) or (5.15) will give $\{p_n(D;j) : D\in\mathcal{S}, j\in D\cap V(n)\}$ or $\{P_n(D;j) :$

$D \in \mathcal{S}, j \in D \cap V(n)\}$, respectively. The existence and uniqueness of the stationary distribution of $Y(t)$ ensures that the system of equations (5.15) possesses a unique solution satisfying $P_n(D;j) \geq 0$, $j \in \mathcal{I}_D$, and $\sum_{j \in \mathcal{I}_D} P_n(D,j) = 1$ (see (5.11)).

For product form protocols for which, as shown in Section 5.1, $U(D) = M(D)$ and $J(D) = D$, equation (5.15) simplifies to

$$\begin{aligned}
&\left[\sum_{k \in D} \mu_k + \sum_{l \in U(D)} \lambda_l\right] P_n(D;j) \\
&\qquad = \left[1 - \sum_{\substack{m \in D-\{j\} \\ m \in V(n)}} P_n(D - \{j\}; m)\right] \mu_j \mathcal{P}_n(D - \{j\}; j) \\
&\qquad + \sum_{k \in D - \{j\}} P_n(D - \{k\}; j)\, \mu_k \\
&\qquad + \sum_{\substack{l \in U(D) \\ l \notin E(n)}} P_n(D \cup \{l\}; j)\, \lambda_l, \qquad D \in \mathcal{S}, \quad j \in D \cap V(n).
\end{aligned} \tag{5.16}$$

Equation (5.13) shows clearly the form of decoupling afforded by the systems under consideration in this Chapter. If we write (5.1) and (5.13) as a single system of linear equations and consider the unknowns grouped in blocks, with the zeroth block consisting of $\{p(D) : D \in \mathcal{S}\}$, and the k-th block, $k = 1, \ldots, N$, consisting of $\{p_{n_k}(D;j) : D \in \mathcal{S},\ j \in D \cap V(n_k)\}$, where $n_1, \ldots, n_N$ are the network nodes, then the coefficient matrix of this system has the structure shown in Figure 5.10. Thus, after the equations for $\{p(D) : D \in \mathcal{S}\}$ are solved, each of the sets of equations for $\{p_{n_k}(D;j) : D \in \mathcal{S},\ j \in D \cap V(n_k)\}$ can be solved independently from each other (and in parallel). Equation (5.16) is a particular case in which the solution of the top block of Figure 5.10 is known in advance, and thus the corresponding unknowns are removed from the subsequent blocks (after a rescaling of the remaining unknowns).

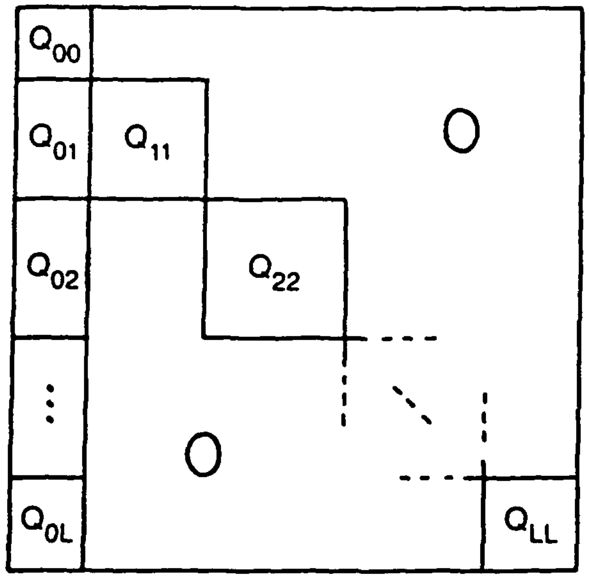


Fig. 5.10 Structure of the matrix of the coefficients of systems obtained from (5.1) and (5.13)

5.3 Throughput Analysis

Given the Markov chains describing the activity of a packet radio network, we wish to find an expression for the throughput of each link as a function of the transition rates and the steady-state probability distribution of the Markov chains. We start in this Section by presenting general expressions for the link throughput of capture modes in class C_s, defined in Section 4.1.3. As will be seen, this analysis only makes use of the representation of the transmitter activity. We then apply these general results to the particular cases of zero capture and perfect capture. We will then give the analysis of the more general capture modes, which requires in addition the description of the receiver activity. We conclude by presenting some analytical examples of application.

By definition, the throughput of link i, S_i, is the long-run fraction of time that link i is engaged in successful transmissions. For the purpose of analysis, we make

the restriction, always satisfied in practice, that the success of a transmission does not depend on the behavior of the system after the termination of the transmission in question.

5.3.1 Capture Modes of Class C_s

Recall that a capture mode is said to belong to class C_s if, for any link j, the availability of the destination receiver of link j to lock onto a packet transmitted on that link is completely determined by the set D of links that are active just before the transmission of the packet starts. Thus for this class of capture modes one does not need to solve equations (5.13) or (5.15) to determine the probability of finding an idle receiver given the set D of active links.

5.3.1.1 General Throughput Equations

Let $X(t)$ represent the set of links which are active at time t, as in Section 5.1, and let $\mathcal{U}(i)$ be the collection of states $D \in \mathcal{S}$ that do not block link i. Define $S(D,i), D \in \mathcal{U}(i)$, to be the fraction of time that link i is engaged in successful transmissions and the state just prior to the start of those transmissions is D. $S(D,i)$ accounts for all successful transmissions on link i that are initiated by a jump of the Markov chain $\{X(t)\}$ from state D into state $D \cup \{i\}$. Summing over D we obtain

$$S_i = \sum_{D \in \mathcal{U}(i)} S(D,i). \tag{5.17}$$

For a fixed D and by the strong Markov property ([Çinl75]), the times of the successive transitions from state D to state $D \cup \{i\}$ are regeneration points for $\{X(t)\}$. We now consider the cycles defined by the time intervals between two successive regeneration points. Let $C_k(D,i)$ denote the length of the k-th cycle, and $\overline{T}_k(D,i)$

be the total time in cycle k that the channel was used successfully by a transmission over link i. We can think of $\overline{T}_k(D,i)$ as the "reward" (for the purpose of calculating the link's throughput) earned during the k-th cycle. With our assumptions on the protocols and modes of operation of the network, $\{(C_k(D,i), T_k(D,i)) : k \geq 1\}$ is a sequence of independent and identically distributed pairs of random variables. In general the elements of each pair are correlated. In the following we will omit the subscripts in these variables whenever we refer to a generic one. If we let $N(t)$ be the number of cycles completed by time t, then

$$S(D,i) = \lim_{t \to \infty} \frac{1}{t} \sum_{k=1}^{N(t)} T_k(D,i).$$

Let $E[C(D,i)]$ and $E[T(D,i)] \equiv \overline{T}(D,i)$ denote the expected cycle length and expected reward, respectively. Standard theorems in the theory of renewal processes ([Ross70]) assert that, with probability one,

$$S(D,i) = \frac{E[T(D,i)]}{E[C(D,i)]}. \tag{5.18}$$

The quantities on the right-hand side of the last equation can be computed in terms of the parameters of the system.

Proposition 5.3.1 *The expected cycle length $E[C(D,i)]$ is given by*

$$E[C(D,i)] = \frac{1}{\lambda_i p(D)} \quad , \quad D \in \mathcal{U}(i). \tag{5.19}$$

Proof: Let $n(t)$ denote the number of state transitions of the Markov chain $X(t)$ in $(0,t]$, and define the embedded Markov chain $\{X_k : k \geq 0\}$ by

$$X_k = X(t), \quad \text{for any } t \text{ such that} \quad n(t) = k\,.$$

X_k is the state the Markov chain $X(t)$ is in after the k-th transition. The irreducibility and positive recurrence of $X(t)$ implies the same properties for the embedded chain X_k, which then possesses a stationary distribution $\{\pi(D) : D \in \mathcal{S}\}$, with

$$\pi(D) = \lim_{k \to \infty} P[X_k = D].$$

The relation between $\pi(D)$ and $Q(D)$ is given by ([Ross70])

$$\frac{p(D)}{p(D')} = \frac{\pi(D)H(D)}{\pi(D')H(D')}, \qquad D, D' \in \mathcal{S} \tag{5.20}$$

where $H(D)$ is the expected sojourn time of $X(t)$ in state D, given by

$$H(D) = \frac{1}{\sum_{D' \in \mathcal{S}} q(D, D')}. \tag{5.21}$$

Given that $X(t)$ is a continuous time Markov chain (and hence the next state after state $D \in \mathcal{S}$ is determined by the minimum of an independent collection of exponential random variables with parameters $q(D, D')$, $D' \in \mathcal{S}$), the transition probabilities for the embedded chain X_k are easily computed; in particular, those relevant to this proof are given by

$$P[X_{k+1} = D \cup \{i\} \mid X_k = D] = \frac{\lambda_i}{\sum_{D' \in \mathcal{S}} q(D, D')}, \qquad D \in \mathcal{U}(i). \tag{5.22}$$

In order to compute the average cycle length we note that ([Ross70])

$$\lim_{t \to \infty} P[X(t) = D \cup \{i\}, X_{n(t)-1} = D] = \frac{H(D \cup \{i\})}{E(C(D, i))}. \tag{5.23}$$

Developing the left-hand side of (5.23) gives us

$$\lim_{t \to \infty} P[X(t) = D \cup \{i\}, X_{n(t)-1} = D]$$

$$= p(D \cup \{i\}) \lim_{t \to \infty} P[X_{n(t)-1} = D \mid X_{n(t)} = D \cup \{i\}]$$

$$= p(D \cup \{i\}) \; P[X_{k+1} = D \cup \{i\} \mid X_k = D] \frac{\pi(D)}{\pi(D \cup \{i\})}$$

which, when substituted in (5.23) after using (5.20)–(5.22), gives Equation (5.19). ∎

From (5.17), (5.18) and Proposition 5.3.1 we then obtain

Proposition 5.3.2 *The throughput of link i, S_i, is given by*

$$S_i = \lambda_i \sum_{D \in \mathcal{U}(i)} p(D)\, \overline{T}(D,i)$$

or, defining the normalized rescheduling rate $G_i \triangleq \frac{\lambda_i}{\mu_i}$,

$$S_i = G_i \sum_{D \in \mathcal{U}(i)} p(D)\, \mu_i \overline{T}(D,i).$$

$\overline{T}(D,i)$ is the product of the probability that a packet is successful, and the average packet length conditioned on it being successful. The probability of the packet being successful is itself the product of the probability $\mathcal{P}_{d(i)}(D,i)$ that the destination receiver will successfully lock onto the packet, the probability that the packet is not lost due to interference within the vulnerable period, if any, and the probability that no bit errors occur, given that the packet is successfully locked onto and no vulnerable period interference occurs. Note that, in order to compute these quantities, one need not assume a particular interference model such as the one of Appendix I, this model being just one case where those computations are straightforward. We now present the computation of $\overline{T}(D,i)$ for two capture modes, zero capture and idealistic perfect capture (see Section 4.1.3 for the definitions), that can be obtained by particularization of the parameters of the capture model of Appendix I.

5.3.1.2 Idealistic Perfect Capture

Let $\mathcal{C}(i)$ be the set of links emanating from i's neighbors. Under idealistic perfect capture a transmission over link i is successful if and only if at the time it starts no other link in the set $\mathcal{C}(i)$ is active, irrespective of what happens after the start of the transmission over link i.

Let $\mathcal{U}_s(i)$ be the subset of $\mathcal{U}(i)$ formed by those states that do not contain any link in $\mathcal{C}(i)$. A receiver will lock onto a new packet with probability 1 if and only if the state just prior to the start of the packet is in $U_s(i)$. For $D \in \mathcal{U}(i) - \mathcal{U}_s(i)$, we have $\overline{T}(D,i) = 0$; for $D \in \mathcal{U}_s(i)$, we have $\overline{T}(D,i) = 1/\mu_i$. We thus have

Proposition 5.3.3 *The throughput of link i under perfect capture is given by*

$$S_i = G_i \sum_{D \in \mathcal{U}_s(i)} p(D). \tag{5.24}$$

Equation (5.24) was first derived in [Boor80] for CSMA, using a heuristic argument.

5.3.1.3 Zero Capture

Under zero capture we assume that a transmission over link i is successful if and only if at the time it starts and during the whole duration of the transmission no link in $\mathcal{C}(i)$ is ever active.

As in the case of perfect capture, for all states D in $\mathcal{U}(i) - \mathcal{U}_s(i)$, $\overline{T}(D,i) = 0$, and thus

$$S_i = G_i \sum_{D \in \mathcal{U}_s(i)} p(D)\ \mu_i \overline{T}(D,i).$$

However, in this case the average transmission time of a successful message is not $1/\mu_i$, due to the dependency that exists between the message length and its success.

The computation of $\overline{T}(D, i)$ involves the construction of an auxiliary Markov chain. In the original chain, let $\mathcal{A}_s(i)$ be the collection of states in which i is active and no element of $\mathcal{C}(i)$ is active, let $\mathcal{A}_c(i)$ be the collection of states in which i is active and some element of $\mathcal{C}(i)$ is active, and let $\mathcal{J}(i)$ be the set of states obtained from $\mathcal{A}_s(i)$ by deactivating link i (see Figure 5.11). With respect to these definitions, the start of a transmission over link i which does not suffer a collision at its very start corresponds to a transition of the Markov chain $X(t)$ from a state $D \in \mathcal{U}_s(i)$ into state $D \cup \{i\} \in \mathcal{A}_s(i)$. $X(t)$ will remain in $\mathcal{A}_s(i)$ as long as i is active and not collided with. A possible later collision of i with a transmission over some other link in $\mathcal{C}(i)$ corresponds to a transition from some state in $\mathcal{A}_s(i)$ into a state in $\mathcal{A}_c(i)$. The successful completion of link i's transmission corresponds to a transition from some state in $\mathcal{A}_s(i)$ into a state in $\mathcal{J}(i)$ without having previously visited any state in $\mathcal{A}_c(i)$.

The structures of $\mathcal{U}_s(i)$ and $\mathcal{A}_s(i)$ are related. Any state of the form $D \cup \{i\}$, with $D \in \mathcal{U}_s(i)$, is in $\mathcal{A}_s(i)$. However, if $X(t)$ is not reversible, $\mathcal{A}_s(i)$ will contain other states. These states are the ones that contain some link $j \notin \mathcal{C}(i)$ that blocks link i but is not blocked by i. Any state containing such a j cannot clearly be of the form $D \cup \{i\}$, with $D \in \mathcal{U}_s(i)$, since then we would have $j \in D$ and thus i would be blocked by D, contrary to the definition of $\mathcal{U}_s(i)$; but nevertheless there will be states in $\mathcal{A}_s(i)$ containing such links j, namely the state $\{i, j\}$. Any state in $\mathcal{A}_s(i)$ not containing any such j will be of the form $D \cup \{i\}$, for some $D \in \mathcal{U}_s(i)$.

The auxiliary Markov chain is now constructed by grouping all states in $\mathcal{J}(i)$ into one absorbing state denoted again $\mathcal{J}(i)$, grouping all states in $\mathcal{A}_c(i)$ into another absorbing state of the same name, and deleting all states not in $\mathcal{A}_s(i) \cup \mathcal{A}_c(i) \cup \mathcal{J}(i)$. When deleting a state, all arrows incident to that state are deleted. In this new chain, the states in $\mathcal{A}_s(i)$ are transient and, with probability 1, $X(t)$ will be absorbed

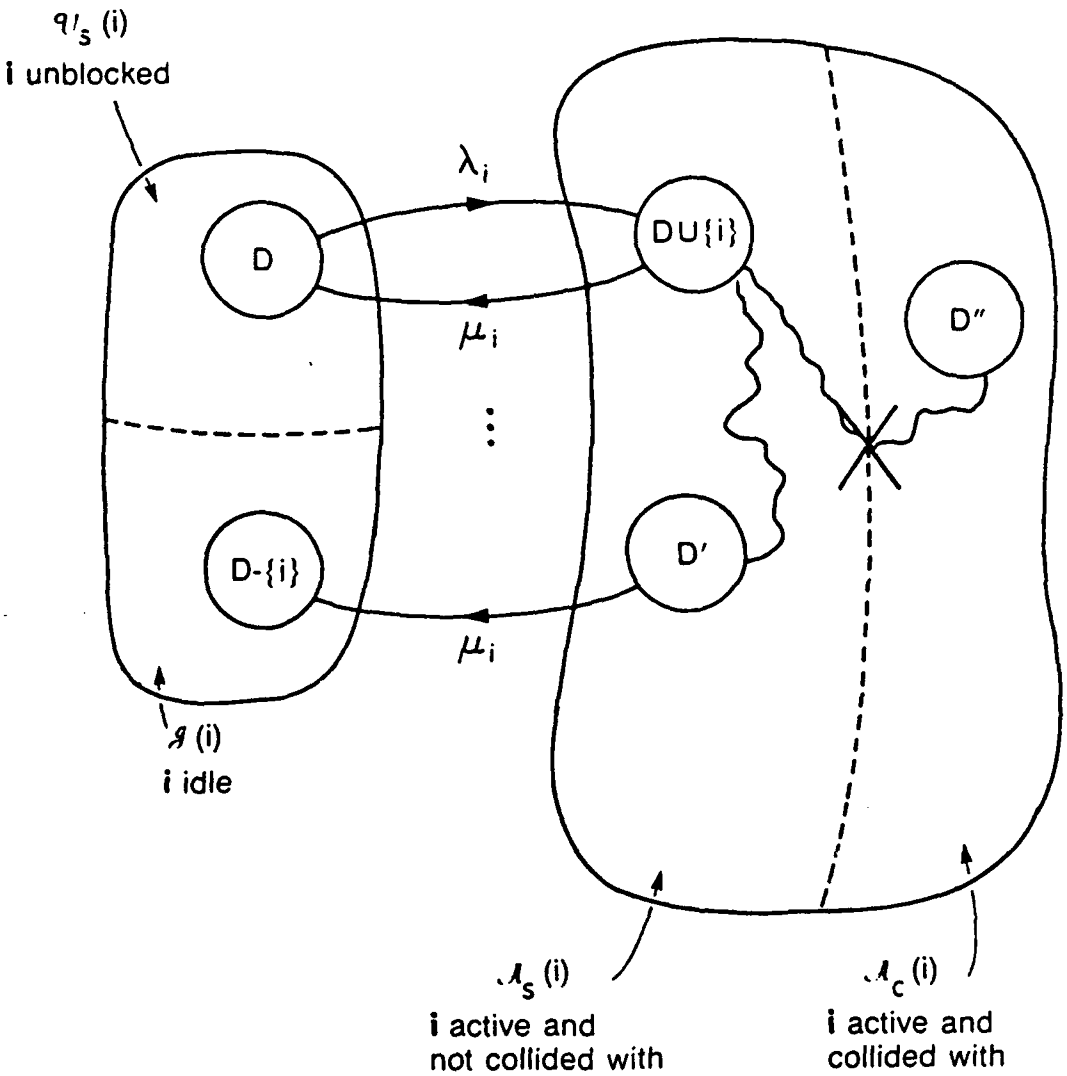


Fig. 5.11 Auxiliary Markov chain for computation of throughput

in either $\mathcal{A}_c(i)$ or $\mathcal{J}(i)$. From what was said above, we see that a transmission over link i, initiated successfully by a jump of $X(t)$ from some $D \in \mathcal{U}_s(i)$ into $D \cup \{i\}$, will terminate successfully if $X(t)$ is absorbed in $\mathcal{J}(i)$, and will suffer a collision if absorption occurs in $\mathcal{A}_c(i)$. Thus, for $D \in \mathcal{U}_s(i)$, $T(D,i)$ equals the length of the time interval between the first entrance to $D \cup \{i\}$ and absorption in $\mathcal{J}(i)$, if absorption occurs in $\mathcal{J}(i)$, and 0, otherwise.

Let k be the cardinality of $\mathcal{A}_s(i)$. By suitable reordering of the states of the modified chain, let its transition rate matrix be

$$\mathbf{R}^*(i) = \begin{bmatrix} \mathbf{R}_s(i) & \mu_i \mathbf{1} & \varphi \\ \mathbf{z} & 0 & 0 \\ \mathbf{z} & 0 & 0 \end{bmatrix} \tag{5.25}$$

where $\mathbf{R}_s(i)$ is the $(k \times k)$ matrix of the transition rates between states in $\mathcal{A}_s(i)$, $\mathbf{e} \triangleq [1 \dots 1]^T$ is of dimension $(k \times 1)$, $\mu_i \mathbf{1}$ is the vector of the transition rates from $\mathcal{A}_s(i)$ into $\mathcal{J}(i)$, φ is the $(k \times 1)$ vector of the transition rates from $\mathcal{A}_s(i)$ into $\mathcal{A}_c(i)$, and $\mathbf{z} = [0 \dots 0]$ is of dimension $(1 \times k)$. With these definitions, we have

Proposition 5.3.4 *The throughput of link i under zero capture is given by*

$$S_i = G_i \sum_{D \in \mathcal{U}_s(i)} p(D)\, \mu_i \overline{\mathbf{T}}_{D \cup \{i\}}, \tag{5.26}$$

where $\overline{\mathbf{T}}_{D\cup\{i\}}$ is the component with index $D \cup \{i\}$ of the column vector

$$\overline{\mathbf{T}} = \mu_i\, \mathbf{R}_s^{-2}(i)\, \mathbf{1}\,.$$

Proof: As we saw above, $\overline{T}(D,i)$ is the average time to absorption in $\mathcal{J}(i)$ for a chain started in state $D \cup \{i\}$, over the set of sample paths for which absorption in $\mathcal{J}(i)$ occurs. Thus $\overline{T}(D,i)$ can be determined from the probability transition matrix, $\mathbf{P}^*(t)$, of the modified chain, defined by

$$\mathbf{P}^*_{D,D'}(t) = P[X(t) = D' \mid X(0) = D]$$

for $D,\ D' \in \mathcal{A}_s(i) \cup \mathcal{A}_c(i) \cup \mathcal{J}(i)$. The transition probability matrix corresponding to $\mathbf{R}^*(i)$ has the form

$$\mathbf{P}^*(t) = \begin{bmatrix} \mathbf{P}_s(t) & \mathbf{P}_{\mathcal{J}}(t) & \mathbf{P}_c(t) \\ \mathbf{z} & 1 & 0 \\ \mathbf{z} & 0 & 1 \end{bmatrix}$$

and is determined by the forward Kolmogorov equation

$$\tfrac{d}{dt}\mathbf{P}^*(t) = \mathbf{P}^*(t) \cdot \mathbf{R}^*(i) \qquad , \quad \mathbf{P}^*(0) = \mathbf{I}.$$

Given the structure of $\mathbf{R}^*(i)$ the forward Kolmogorov equation takes the form

$$\begin{aligned} \tfrac{d}{dt}\mathbf{P}_s(t) &= \mathbf{P}_s(t)\ \mathbf{R}_s(i) \qquad &, \quad \mathbf{P}_s(0) = \mathbf{I} \\ \tfrac{d}{dt}\mathbf{P}_{\mathcal{J}}(t) &= \mu_i\ \mathbf{P}_s(t)\,\mathbf{1} \qquad &, \quad \mathbf{P}_{\mathcal{J}}(0) = \mathbf{0} \\ \tfrac{d}{dt}\mathbf{P}_c(t) &= \mathbf{P}_s(t)\,\varphi \qquad &, \quad \mathbf{P}_c(0) = \mathbf{0} \end{aligned}$$

with solution

$$\begin{aligned} \mathbf{P}_s(t) &= e^{\mathbf{R}_s(i)t} \qquad &, \quad t \geq 0 \\ \mathbf{P}_{\mathcal{J}}(t) &= \mu_i \left(e^{\mathbf{R}_s(i)t} - \mathbf{I}\right) \mathbf{R}_s^{-1}(i)\,\mathbf{1} \quad &, \quad t \geq 0 \\ \mathbf{P}_c(t) &= \left(e^{\mathbf{R}_s(i)t} - \mathbf{I}\right) \mathbf{R}_s^{-1}(i)\,\varphi \quad &, \quad t \geq 0. \end{aligned}$$

Note that, since the states in $\mathcal{A}_s(i)$ are transient, $e^{\mathbf{R}_s(i)t} \to 0$ as $t \to \infty$. Let now $\mathbf{T}$ be a column vector with rows indexed by the states in $\mathcal{A}_s(i)$ in the same order as the rows of $\mathbf{R}_s(i)$ and where, for $D' \in \mathcal{A}_s(i)$, the component with index D', $\mathbf{T}_{D'}$, is the random variable giving the *time to absorption in* $\mathcal{J}(i)$ when the chain is started in state D'. Then

$$P\{\mathbf{T} \leq t\ \mathbf{1}\} = P_{\mathcal{J}}(t) \quad , \qquad t \geq 0$$

and

$$\overline{\mathbf{T}} \triangleq E\{\mathbf{T}; \mathbf{T} < \infty\} = \int_0^\infty [\mathbf{P}_{\mathcal{J}}(\infty) - \mathbf{P}_{\mathcal{J}}(t)]\,dt$$

$$= -\mu_i \int_0^\infty e^{\mathbf{R}_s(i)t}\,\mathbf{R}_s^{-1}(i)\,\mathbf{1}\,dt = \mu_i\ \mathbf{R}_s^{-2}(i)\,\mathbf{1}.$$

Since $\overline{T}(D,i) = \overline{\mathbf{T}}_{D\cup\{i\}}$, we obtain equation (5.26). ∎

Proposition 5.3.5 *The probability of success for a packet of link i whose transmission starts with the system on state D is given by the component with index $D \cup \{i\}$ of the column vector*

$$\mathbf{P}_s = -\mu_i\,\mathbf{R}_s^{-1}(i)\,\mathbf{1}.$$

Proof: The probability of successful transmission is the probability that the auxiliary Markov chain is absorbed in $\mathcal{J}(i)$. From the proof of Proposition 5.3.4, it is given by

$$\lim_{t\to\infty} \mathbf{P}_{\mathcal{J}}(t) = -\mu_i\,\mathbf{R}_s^{-1}(i)\,\mathbf{1}.$$

∎

Given the column vectors $A = [a_i]$ and $B = [b_i]$, let $C = A \oslash B$ and $D = A \odot B$ denote the column vector $C = [c_i]$ such that $c_i = a_i/b_i$ and $D = [d_i]$ such that $d_i = a_i b_i$, respectively. (The vector $A \odot B$ is often called the *Schur product* of A and B.)

Proposition 5.3.6 *The average length of a successful packet sent over link i whose transmission starts with the system in state D is given by the component with index $D \cup \{i\}$ of the column vector*

$$\mathbf{L}_s = [\mu_i\,\mathbf{R}_s^{-2}(i)\,\mathbf{1}] \oslash [-\mu_i\,\mathbf{R}_s^{-1}(i)\,\mathbf{1}] \tag{5.27}$$

$$= \frac{1}{\mu_i}\,\mathbf{1} - [\mathbf{R}_s^{-2}(i)\,\varphi] \oslash [\mathbf{1} + \mathbf{R}_s^{-1}(i)\,\varphi]. \tag{5.28}$$

Proof: Equation (5.27) derives directly from Propositions 5.3.4 and 5.3.5. For Equation (5.28), note that the rows of $\mathbf{R}(i)$ add up to zero, so that

$$\mathbf{R}_s(i)\,\mathbf{1} + \mu_i\,\mathbf{1} + \varphi = \mathbf{0}\,.$$

Multiplying by $\mathbf{R_s}^{-1}(i)$ and $\mathbf{R_s}^{-2}(i)$ and rearranging we obtain, respectively,

$$-\mu_i\,\mathbf{R_s}^{-1}(i)\,\mathbf{1} = \mathbf{1} + \mathbf{R_s}^{-1}(i)\,\varphi$$

and

$$\mu_i\,\mathbf{R_s}^{-2}(i)\,\mathbf{1} = -\mathbf{R_s}^{-1}\,\mathbf{1} - \mathbf{R_s}^{-2}(i)\,\varphi.$$

Substituting these expressions in (5.27), we obtain (5.28). ∎

Proposition 5.3.7 *The average length of an unsuccessful packet of link i whose transmission starts with the system in state D is given by the component with index $D \cup \{i\}$ of the column vector*

$$\mathbf{L}_u = \frac{1}{\mu_i}\,\mathbf{1} + [\mathbf{R_s}^{-2}(i)\,\varphi] \oslash [-\mathbf{R_s}^{-1}(i)\,\varphi].$$

Proof: The vectors of the average lengths of the channel packets, the successful packets, and the unsuccessful packets, are related by

$$\frac{1}{\mu_i}\,\mathbf{1} = \mathbf{L}_s \odot \mathbf{P}_s + \mathbf{L}_u \odot (\mathbf{1} - \mathbf{P}_s)$$

from which

$$\mathbf{L}_u = \frac{\frac{1}{\mu_i}\,\mathbf{1} - \mathbf{L}_s \odot \mathbf{P}_s}{\mathbf{1} - \mathbf{P}_s}. \tag{5.29}$$

But

$$\mathbf{1} - \mathbf{P}_s = \mathbf{1} + \mu_i\,\mathbf{R_s}^{-1}\,\mathbf{1} = -\mathbf{R_s}^{-1}\,\varphi \tag{5.30}$$

and

$$
\begin{aligned}
\frac{1}{\mu_i}\mathbf{1} - \mathbf{L}_s \odot \mathbf{P}_s &= \frac{1}{\mu_i}\mathbf{1} - \mu_i\, \mathbf{R}_s^{-2}(i)\,\mathbf{1} \\
&= \frac{1}{\mu_i}\mathbf{1} + \mathbf{R}_s^{-1}(i)\,\mathbf{1} + \mathbf{R}_s^{-2}(i)\,\varphi \\
&= \frac{1}{\mu_i}\left[\mathbf{1} + \mu_i\, \mathbf{R}_s^{-1}(i)\,\mathbf{1}\right] + \mathbf{R}_s^{-2}(i)\,\varphi \\
&= -\frac{1}{\mu_i}\, \mathbf{R}_s{}^{-1}(i)\,\varphi + \mathbf{R}_s^{-2}(i)\,\varphi.
\end{aligned}
\tag{5.31}
$$

Combining (5.29)-(5.31) we obtain the desired result. ∎

Proposition 5.3.8 *If the rates of transition from state $D \in \mathcal{A}_s(i)$ into $\mathcal{A}_c(i)$ are independent of D, the successful packets have an exponential length distribution.*

Proof: The distribution function vector of the successful packets is given by

$$
\begin{aligned}
\mathbf{F}_s(t) &= \mathbf{P}_{\mathcal{J}}(t) \oslash \left[-\mu_i\, \mathbf{R}_s^{-1}(i)\,\mathbf{1}\right] \\
&= \left[\mu_i\, \mathbf{R}_s^{-1}(i)\left(e^{\mathbf{R}_s(i)t} - \mathbf{I}\right)\mathbf{1}\right] \oslash \left[-\mu_i\, \mathbf{R}_s^{-1}(i)\,\mathbf{1}\right].
\end{aligned}
$$

From the hypothesis of the Proposition, $\varphi = \lambda_c \mathbf{1}$, for some $\lambda_c > 0$. Then

$$
\mathbf{R}_s(i)\,\mathbf{1} + \mu_i\,\mathbf{1} + \lambda_c\,\mathbf{1} = \mathbf{0}
$$

or

$$
\mathbf{R}_s(i)\,\mathbf{1} = -(\mu_i + \lambda_c)\,\mathbf{1}\,,
$$

which shows that $\mathbf{1}$ is an eigenvector of $\mathbf{R}_s(i)$ with associated eigenvalue $-(\mu_i + \lambda_c)$. It then follows that

$$
\mathbf{R}_s^{-1}(i)\,\mathbf{1} = -(\mu_i + \lambda_c)^{-1}\mathbf{1}
$$

and

$$e^{\mathbf{R}_s(i)t}\,\mathbf{1} = e^{-(\mu_i+\lambda_c)t}\,\mathbf{1}$$

so that

$$\mathbf{F}_s(t) = \left[1 - e^{-(\mu_i+\lambda_c)t}\right]\mathbf{1}.$$

∎

A situation in which the hypothesis of Proposition 5.3.8 is satisfied is ALOHA with zero capture. Indeed, in this case, no link emanating from any neighbor of i's destination is active in $A_s(i)$, and the rate of transition from any $D \in A_s(i)$ into $A_c(i)$ is the rate at which any of these links becomes active, thus independent of D. The proof of Proposition 5.3.8 also gives the average length of the successful packets, and the probability of success for a packet.

5.3.2 General Capture Modes

In the case of capture modes in class $\mathcal{C}_s$, the availability of an idle receiver to lock onto a new packet could be readily deduced by examining the set D of links active just before the start of the packet transmission. In more general capture modes, such as those of typical spread spectrum systems, the situation is different. Due to the orthogonality properties of the codes used, a receiver which is not locked onto a packet is able to lock onto a new packet even in the presence of remainders of neighboring transmissions that were not locked onto to. Thus, in order to determine the success or failure of a packet, we must not only know which links are active at the time its transmission starts, but also whether the destination receiver is free to lock onto the new packet. Consequently, for the computation of the throughput of a given link we need to consider the joint description of the activity of the

transmitters and the activity of the receiver at the destination node of the link described in Section 5.2. To these processes we then apply the methods of Section 5.3.1.

5.3.2.1 General Throughput Equations

Consider that we are interested in the throughput S_i of link i, and let $d(i)$ be its destination node. Again we can write

$$S_i = \sum_{D \in \mathcal{U}(i)} S(D,i), \tag{5.32}$$

where $S(D,i)$ is the throughput corresponding to the successful transmissions of link i which find the set of active links just prior to their start to be D. We now examine the process $Y_n(t)$ to determine these partial throughputs. The only transitions which contribute to the throughput $S(D,i)$ are those from state $(D;0)$ into state $(D \cup \{i\}; i)$. We use these transitions to define regenerative cycles, for which we again let $C(D,i)$ denote the length of a typical cycle, and $T(D,i)$ the total time during a typical cycle that the channel is used by a successful transmission over link i. The "partial" throughput $S(D,i)$ is given by

$$S(D,i) = \frac{E[T(D,i)]}{E[C(D,i)]}, \tag{5.33}$$

with (cf. Proposition 5.3.1)

$$\begin{aligned} E[C(D,i)] &= \frac{1}{q_{d(i)}\left((D;0),(D\cup\{i\};i)\right)\, p_{d(i)}(D;0)} \\ &= \frac{1}{\lambda_i\, \mathcal{P}_{d(i)}(D;i)\, p_{d(i)}(D;0)}. \end{aligned} \tag{5.34}$$

Note that here $T(D,i)$ denotes the time that the channel is successfully used by i's packet given that the packet was locked onto by the destination, whereas

in Section 5.3.1 it denoted the time that the channel is successfully used by i's packet given that i became active. $E[T(D,i)]$ as defined in Section 5.3.1 is thus the product of $\mathcal{P}_{d(i)}(D,i)$ and $E[T(D,i)]$ as defined here. $E[T(D,i)]$ depends on the conditions under which the packet will be successfully received after the preamble is successfully locked onto.

Combining Equations (5.32)–(5.34) we have

Proposition 5.3.9 *The throughput of link i is given by*

$$S_i = \lambda_i \sum_{D \in \mathcal{U}(i)} p_{d(i)}(D;0)\, \mathcal{P}_{d(i)}(D;i)\, E[T(D,i)] \tag{5.35}$$

or, in terms of the conditional probabilities $P_{d(i)}(D,\varphi)$ introduced in (5.14),

$$S_i = \lambda_i \sum_{D \in \mathcal{U}(i)} p(D)\, P_{d(i)}(D;0)\, \mathcal{P}_{d(i)}(D;i)\, E[T(D,i)]. \tag{5.36}$$

5.3.2.2 Spread Spectrum Capture

We discuss now the computation of $E[T(D,i)]$ for the model for correct packet reception introduced in Appendix I. The capture mode obtained from that model is defined by the following conditions, concerning the correct reception of link i's packet:

(i) given that the set of links active just before the start of i's transmission is D, the packet is lost due to interference by transmissions with the same code sequence and whose timing falls within i's packet vulnerable period with probability $p_{vp}(D,i)$;

(ii) while the packet is being transmitted, it is lost with probability $p_l(i,j)$ whenever link j becomes active, due to j's timing falling within i's vulnerable

window, or because link j's source node coincides with link i's destination node (this last case occurring only for undisciplined protocols, in which case $p_l(i,j)$ equals one);

(iii) after the packet transmission starts, bit errors occur as a Poisson process with rate $\epsilon(D,i)$ whenever the set of active links is D; the packet is lost if at least one bit error occurs.

Remark 5.3.10 The above model for correct reception is a direct extension of the zero capture model of Section 5.3.1.2 to the situation where a vulnerable period and state-dependent bit errors exist. State-dependent bit errors were first considered in the context of a similar analytic framework by Storey and Tobagi in [Stor84].

$E[T(D,i)]$ is computed by "tracking" the behavior of the system since i's transmission starts until it either finishes successfully or the packet is lost due to any of the mechanisms in (i)–(iii) above. (This is the same approach as the one followed in Section 5.3.1.3 for the analysis of zero capture.) Since the loss of the packet after it is locked onto is a function exclusively of the activity of the network transmitters, we only need to consider an auxiliary Markov chain $\{X_j^a(t)\}$ obtained as an absorbing modification of the process $\{X(t)\}$ of the active links.

Let us first assume that no loss occurs when i's packet starts due to correlated interference within the vulnerable period by transmissions with the same code sequence or due to the unavailability of a receiver. Let $\mathcal{A}_s(i)$ be the set of states $D \in S$ such that $i \in D$. After i's packet is successfully locked onto by its destination, the Markov chain $\{X_i^a(t)\}$ will "wander" in $\mathcal{A}_s(i)$ as long as no bit errors or vulnerable window collisions take place. Once one of these events occurs, $\{X_i^a(t)\}$ transits into an absorbing state $\mathcal{A}_c(i)$. While in state $D \in \mathcal{A}_s(i)$, transitions into this absorbing state due to bit errors occur with rate $\epsilon(D,i)$, and transitions due to the start of a new transmission over link j whose timing falls within i's vulnerable window oc-

cur with rate $p_l(i,j)\,\lambda j$ for any $j \in U(D)$. If i's transmission ends successfully, the Markov chain will be absorbed into state $\mathcal{J}(i)$. Within $\mathcal{A}_s(i)$, $\{X_i^a(t)\}$ transits from $D \in \mathcal{A}_s(i)$ into $D \cup \{j\} \in \mathcal{A}_s(i)$ with rate $[1-p_l(i,j)]\lambda j$, and into $D - \{l\} \in \mathcal{A}_s(i)$, $l \in D$, with rate μ_l. $E[T(D,i)]$ is then the average time to absorption in $\mathcal{J}(i)$ over the set of sample paths that terminate in $\mathcal{J}(i)$, given that $\{X_i^a(t)\}$ started in $D \cup \{i\} \in \mathcal{A}_s(i)$. The transition rate matrix $\mathbf{R}^*(i)$ of $\{X_i^a(t)\}$ has the form given in Equation (5.25), and $E[T(D,i)]$ is given by Proposition 5.3.4.

If we now consider the possibility of loss of i's packet at the start of its transmission due to an existing transmission having timing within i's vulnerable window, we have

Proposition 5.3.11 *Under the model for correct reception of Section 5.3.2.2, the average time the channel is successfully used by i's packet in a cycle, given that the packet is successfully locked onto by its destination, is given by*

$$E[T(D,i)] = [1 - p_{vp}(D,i)]\,\overline{\mathbf{T}}_{D\cup\{i\}},$$

where $\overline{\mathbf{T}}_{D\cup\{i\}}$ is the component of index $D \cup \{i\}$ of the column vector

$$\overline{\mathbf{T}} = \mu_i\,\mathbf{R}_s^{-2}(i)\,\mathbf{1}, \tag{5.37}$$

and $\mathbf{R}_s(i)$ is the submatrix of the state transition rate matrix of $\{X_i^a(t)\}$ whose rows and columns correspond to the states in $\mathcal{A}_s(i)$.

Results similar to Propositions 5.3.5–5.3.8 can also be derived for the capture model being considered.

5.3.2.3 Example of Application

In order to illustrate the application of the theory developed in this chapter, we give here an example of the actual computation of the throughput equations.

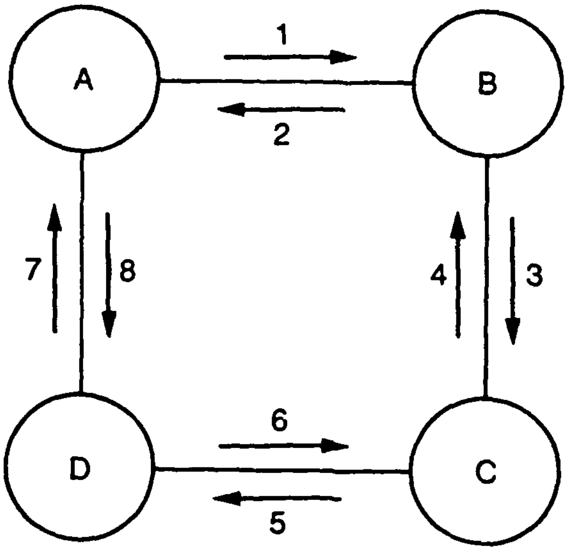


Fig. 5.12 A four node ring network

We consider a four node ring network, shown in Figure 5.12, operating under CSMA and in which receiver-directed bit-homogeneous codes are used. The preamble codes and data portion codes are distinct and, for analytical simplicity, assumed perfectly orthogonal, in such a way that an unlocked receiver locks with probability 1 onto a new packet destined to it. In terms of the parameters defining the system, we have $\mathcal{P}_{d(i)}(D; i) = 1$ if $i \notin D$, and $\mathcal{P}_{d(i)}(D, i) = 0$ otherwise. Also for analytical simplicity we assume that there exists no interference between two transmissions with different codes, which in this case are any two transmissions with different intended destinations. We assume that transmissions with the same destination can interfere if their timing is within the duration of the vulnerable period, whose relative width we designate by p_L, but do not otherwise cause other bit errors. We assume the timing of each interfering signal to be independently and uniformly distributed within a bit duration of the desired signal, so that, in the notation of Section 5.3.2.2, $p_{vp}(D, i) = 1 - (1 - p_L)^{\#_i(D)}$, where $\#_i(D)$ is the number of links in D with the same destination (and hence code sequence) as link i. For the same

reason, $p_l(i,j) = p_L$ if i and j have the same destination, and $p_l(i,j) = 0$ otherwise. Finally, from the assumptions above, $\varepsilon(D,i) \equiv 0$. We only consider the throughput equation for link 1, since all the other ones can be easily obtained by symmetry considerations.

Link 1 is only allowed to transmit in states ϕ, $\{4\}$, and $\{5\}$. Thus from Equations (5.36) and (5.37) we have

$$\begin{aligned}\frac{S_1}{\lambda_1} =& p(\phi)\, P_B\big(\phi;0\big)\, \mathcal{P}_B\big(\phi;1\big)\, \big[1 - p_{vp}\big(\phi,1\big)\big]\, \overline{T}_n\big(\phi,1\big) \\ &+ p\big(\{4\}\big)\, P_B\big(\{4\};0\big)\, \mathcal{P}_B\big(\{4\};1\big)\, \big[1 - p_{vp}\big(\{4\},1\big)\big]\, \overline{T}_n\big(\{4\}\big) \\ &+ p\big(\{5\}\big)\, P_B\big(\{5\};0\big)\, \mathcal{P}_B\big(\{5\};1\big)\, \big[1 - p_{vp}\big(\{5\},1\big)\big]\, \overline{T}_n\big(\{5\},1\big),\end{aligned} \tag{5.38}$$

where $\overline{T}_n(D,i)$ is the component of index $D \cup \{i\}$ of the vector $\mathbf{T}$ given by Equation (5.37). From our assumptions, we have $\mathcal{P}_B(\phi;1) = \mathcal{P}_B(\{4\};1) = \mathcal{P}_B(\{5\};1) = 1$. We also have $P_B(\phi;0) = P_B(\{5\};0) = 1$, $p_{vp}(\phi,1) = p_{vp}(\{5\},1) = 0$, and $p_{vp}(\{4\},1) = p_L$. CSMA with symmetric hearing possesses a product form solution for the steady-state probabilities $\{p(D)\}$, and thus $p(\{4\}) = (\lambda_4/\mu_4)p(\phi)$ and $p(\{5\}) = (\lambda_5/\mu_5)p(\phi)$. Equation (5.38) then becomes

$$\frac{S_1}{\lambda_1} = p(\phi)\left\{\overline{T}_n\big(\phi,1\big) + \frac{\lambda_4}{\mu_4} P_B\big(\{4\},0\big)\,(1-p_L)\,\overline{T}_n\big(\{4\},1\big) + \frac{\lambda_5}{\mu_5}\overline{T}_n\big(\{5\},1\big)\right\}. \tag{5.39}$$

The probability of finding an idle receiver at node B is obtained from the system

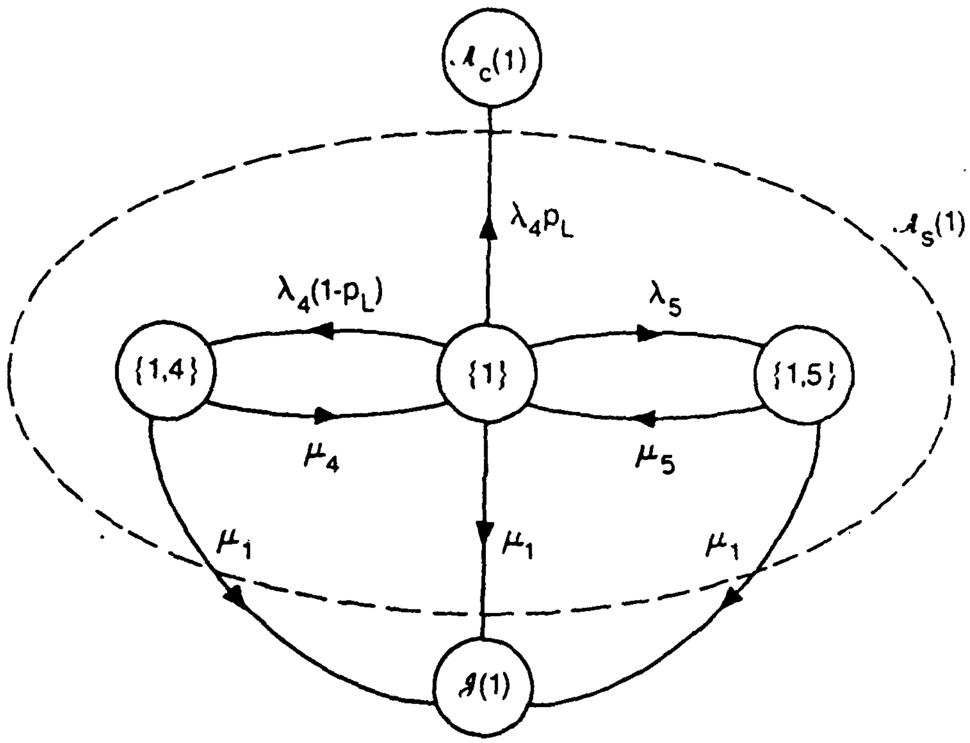


Fig. 5.13 Auxiliary Markov chain for a four node ring

of equations (5.16), which becomes

$$
\begin{aligned}
(\mu_1 + \lambda_4 + \lambda_5)\, P_B\big(\{1\},1\big) &= \mu_1 + \lambda_4\, P_B\big(\{1,4\},1\big) + \lambda_5\, P_B\big(\{1,5\},1\big) \\
(\mu_4 + \lambda_1 + \lambda_8)\, P_B\big(\{4\},4\big) &= \mu_4 + \lambda_1\, P_B\big(\{1,4\},4\big) + \lambda_8\, P_B\big(\{4,8\},4\big) \\
(\mu_1 + \mu_4)\, P_B\big(\{1,4\},1\big) &= \mu_1 \Big[1 - P_B\big(\{4\},4\big)\Big] + \mu_4\, P_B\big(\{1\},1\big) \\
(\mu_1 + \mu_4)\, P_B\big(\{1,4\},4\big) &= \mu_4 \Big[1 - P_B\big(\{1\},1\big)\Big] + \mu_1\, P_B\big(\{4\},4\big) \\
(\mu_1 + \mu_5)\, P_B\big(\{1,5\},1\big) &= \mu_1 + \mu_5\, P_B\big(\{1\},1\big) \\
(\mu_4 + \mu_8)\, P_B\big(\{4,8\},4\big) &= \mu_4 + \mu_8\, P_B\big(\{4\},4\big)
\end{aligned}
\tag{5.40}
$$

The average successful channel utilizations $\overline{T}_n(\cdot,1)$ are obtained from the corresponding auxiliary Markov chain, shown in Figure 5.13, via Equation (5.37). The

matrix $\mathbf{R}_s(1)$ is given by

$$
\mathbf{R}_s(i) = \begin{array}{c} \\ \{1\} \\ \{1,4\} \\ \{1,5\} \end{array}
\begin{array}{c}
\begin{array}{ccc} \{1\} & \{1,4\} & \{1,5\} \end{array} \\
\begin{pmatrix}
-(\mu_1+\lambda_4+\lambda_5) & \lambda_4(1-p_L) & \lambda_5 \\
\mu_4 & -(\mu_1+\mu_4) & 0 \\
\mu_5 & 0 & -(\mu_1+\mu_5)
\end{pmatrix}
\end{array}. \tag{5.41}
$$

Taking, for analytical simplicity, $\mu_, = \cdots = \mu_8 = 1$, we get from (5.40)

$$
P_B(\{4\},0) = 1 - P_B(\{4\},4) = \frac{\frac{\lambda_1}{2}\left(1+\frac{\lambda_5}{2}\right)}{(1+\frac{\lambda_5}{2})(1+\frac{\lambda_1+\lambda_8}{2}) + \frac{\lambda_4}{2}\frac{1+\lambda_8}{2}} \tag{5.42}
$$

and from Equations (5.37) and (5.41)

$$
\begin{aligned}
\overline{T}_n\big(\phi,1\big) &= \frac{(2+\lambda_4+\lambda_5)^2 - \lambda_4(4+\lambda_4+\lambda_5)p_L}{\big[(2+\lambda_4+\lambda_5)+\lambda_4 p_L\big]^2} \\
\overline{T}_n(\{4\},1) = \overline{T}_n(\{5\},1) &= \frac{(2+\lambda_4+\lambda_5)^2 - \lambda_4 p_L}{[(2+\lambda_4+\lambda_5)+\lambda_4 p_L]^2}.
\end{aligned} \tag{5.43}
$$

The normalizing constant $p(\phi)$ for the product form solution is given by

$$
p(\phi) = [(1+G_1+G_8)(1+G_4+G_5) + (1+G_2+G_3)(1+G_6+G_7) - 1]^{-1}, \tag{5.44}
$$

where $G_i \triangleq \lambda_i/\mu_i$. Introducing (5.42)-(5.44) in (5.39) we finally obtain the throughput expression for link 1.

5.4 Summary and Conclusions

We presented in this Chapter the analysis of the class of protocols for which the protocol decisions are completely specified by the state of the network transmitters. The protocols in this class lead to decoupling between the description of

the activity of the receivers and the activity of the transmitters, and we refer to the corresponding systems as *decoupled systems*. We described in Section 5.1 a Markovian model for the representation of the transmitter activity. The states of this process are sets of active links. The corresponding state space was characterized as being formed by those states that can be reached from the idle state by means of link activations only. We then wrote the equilibrium equations for the steady-state probability distribution of this process, and investigated the conditions under which this distribution possesses a product form solution. The existence of a product form solution was shown to be equivalent to reversibility of the stochastic process, and also to be equivalent to symmetry in the blocking between any two pairs of links. This last conditions provides a criterion for the existence of a product form that is easy to verify from the specification of the channel access protocol. We also showed that, despite its analytical simplicity, the computation of the normalization factor appearing in the product form solution is NP-hard.

In Section 5.2 we presented a Markovian model for the description of the activity of the receivers. For this model we derived the equilibrium equations of the corresponding stochastic process in a form that emphasizes the decoupling afforded by the class of protocols considered in this Chapter.

Section 5.3 presented the derivation of throughput measures from (i) the steady-state probabilities of the processes describing the activity of the transmitter and receivers, (ii) the probability of a packet being successful, and (iii) the average length of the successful packets. For the capture modes considered, the two last quantities are computed by means of absorbing modifications of the Markov processes involved. This Section also presented expressions for the probability of packet success, for the average length of the successful packets, and for the average length of the unsuccessful packets. Finally, the derivation of the throughput equations for a

four-node ring network was given, as an example of application of the formalism developed in this Chapter.

Chapter 6

COUPLED SYSTEMS

We present in this Chapter the analysis of the more general class of channel access protocols that do not lead to decoupling between the descriptions of the activity of the transmitters and the activity of the receivers (i.e., that do not belong to class $\mathcal{D}$. In Section 6.1 we present a Markovian model for the description of the network activity. We give the characterization of the state space of the associated stochastic process, and derive the balance equations satisfied by its steady-state probability distribution. In Section 6.2 we present the derivation of throughput equations, using an approach similar to that of Chapter 5.

6.1 Description of Network Activity

We introduced in Section 4.1.2 a network state description sufficient for the purposes of implementation of the channel access protocol. This description contains, for each node, the information of whether the node is transmitting, locked onto a packet, or otherwise idle. For notational convenience, we shall use here a state description that, while carrying the same information, is in a form closer to that of Chapter 5.

6.1.1 State Description and Definitions

Since we consider that only the destination node of a link can lock onto the link's transmissions, the state of the nodes of the network can be unambiguously defined by adding to the state description of Chapter 5 the specification of which active links are locked onto by their destination nodes, and which are not. We thus represent the state of the network by $D = (D^+; D^-)$, where D^+ is the set of links that are *active* and *locked onto* by their destinations, and D^- is the set of links that are *active* and *not locked onto* by their destinations. Given state D and link $j \in \mathcal{L}$, we say that $j \in D$ if either $j \in D^+$ or $j \in D^-$. When notationally convenient, we shall write $j^+ \in D$ to mean $j \in D^+$ and $j^- \in D$ to mean $j \in D^-$.

Definition 6.1.1 Given a network node n, we denote by $E(n)$ the set of all links whose source node is n and by $V(n)$ the set of all links whose destination node is n. Given a link j we denote by $s(j)$ its source node, and by $d(j)$ its destination node.

Definition 6.1.2 Given a state D, we designate by $L_d(D)$ the set of links j such that no link $i \in E(d(j))$ is in D. In words, $L_d(D)$ is the set of links j whose destination nodes are not active in D.

Definition 6.1.3 Given a state D, we designate by $L_s(D)$ the set of links j such that no link $i \in V(s(j))$ is in D^+. In words, $L_s(D)$ is the set of links j whose source nodes are not locked onto some transmission in state D.

Definition 6.1.4a Let $j \in \mathcal{L}$ and D be a state such that $j \notin D$ and $j \in L_s(D)$. We define $D+j^- \triangleq (D^+; D^- \cup \{j\})$. If $j \in L_d(D)$, we define $D+j^+ \triangleq (D^+ \cup \{j\}; D^-)$.

Definition 6.1.4b Let $j \in \mathcal{L}$ and D be a state such that $j \notin D$ and $j \notin L_s(D)$. Let $l \in D^+ \cap V(s(j))$. We define $D + j^- \triangleq (D^+ - \{l\}; D^- \cup \{j\} \cup \{l\})$. If $j \in L_d(D)$, we define $D + j^+ \triangleq (D^+ \cup \{j\} - \{l\}; D^- \cup \{l\})$.

Definition 6.1.5 Let $j \in \mathcal{L}$ and D be a state such that $j \in D$. We define $D - j \triangleq (D^+ - \{j\}; D^-)$ if $j \in D^+$, and $D - j \triangleq (D^+; D^- - \{j\})$ if $j \in D^-$.

Given the assumptions of exponential packet length and scheduling delay distributions of Section 4.2, and of independent locking onto new packets by an idle receiver of Section 4.1.3, it follows that the process $\{X(t) : t \geq 0\}$, defined as $X(t) = D$ if the state of the network at time t is D, is Markovian. We define in the next Section the state space of this process.

6.1.2 State Space

Definition 6.2.1 We say that state $D = (D^+; D^-)$ is *admissible* if (i) for every pair of links $i, j \in D$ we have $s(i) \neq s(j)$, (ii) for every pair of links $i, j \in D^+$ we have $d(i) \neq d(j)$, and (iii) for every pair of links $i \in D^+$ and $j \in D$ we have $d(i) \neq s(j)$. These conditions correspond to the requirements that (i) there is only one transmitter per node, (ii) there is only one receiver per node, and (iii) a node cannot transmit and receive simultaneously.

Definition 6.2.2 We say that link i blocks link j if whenever i is active j is not allowed to transmit by the access protocol. With the notation of Section 4.1.2,

i blocks j if $i \in \mathcal{B}_a(j)$. We say that link i^+ blocks link j if, whenever i is active and locked onto by its destination node, j is not allowed to transmit, but is otherwise allowed to if i is active but not locked onto. In the notation of Section 4.1.2, i^+ blocks j if $i \in \mathcal{B}_l(j)$.

Definition 6.2.3 State $D_1 = (D_1^+; D_1^-)$ is said to be ***protocol-reachable***, or ***p-reachable*** for short, from admissible state $D_2 = (D_2^+; D_2^-)$ if (i) for some link $j \in D_2$ we have $D_1 = D_2 - j$, (ii) for some link such that $j \notin D_2$ and j not blocked by D_2 we have $D_1 = D_2 + j^-$, or (iii) for some link j such that $j \notin D_2, j \in L_d(D)$, and j not blocked by D_2 we have $D_1 = D_2 + j^+$. In words, D_1 is *p-reachable* from state D_2 if D_1 can result from D_2 either by the deactivation of a link active in state D_2, or by the activation of some link not in D_2, according to the rules set by the access protocol.

Definition 6.2.4 A state D is said to be *p-reachable* if there exists a sequence of admissible states $D_0 = \phi, D_1, \cdots, D_q = D$ such that D_k is p-reachable from D_{k-1} for $k = 1, \cdots, q$.

Definition 6.2.5 The state space S is formed by all p-reachable states.

Remark 6.2.6 Any of the operations defined in Definitions 6.1.4 and 6.1.5 when performed over an admissible state produces an admissible state. Since ϕ is admissible, any p-reachable state is also admissible.

Remark 6.2.7 Depending on the particular values of the locking probabilities, one of the transitions from D into $D + j^+$ or $D + j^-$ may have zero probability of occurrence, with the result that the corresponding destination state may be transient. This point is illustrated by the two-node network of Figure 6.1, whose state transition rate diagram under disciplined ALOHA and perfect capture is shown in Figure 6.2. In the situation considered, states $(\phi; \{1\})$ and $(\phi; \{2\})$ have zero probability of being entered from state ϕ, and are transient. Note however that

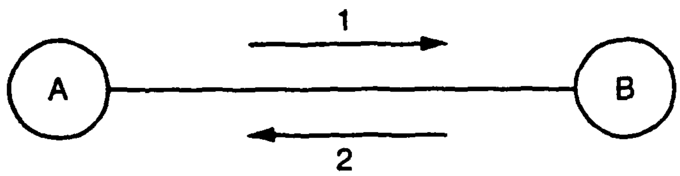


Fig. 6.1 A two-node network

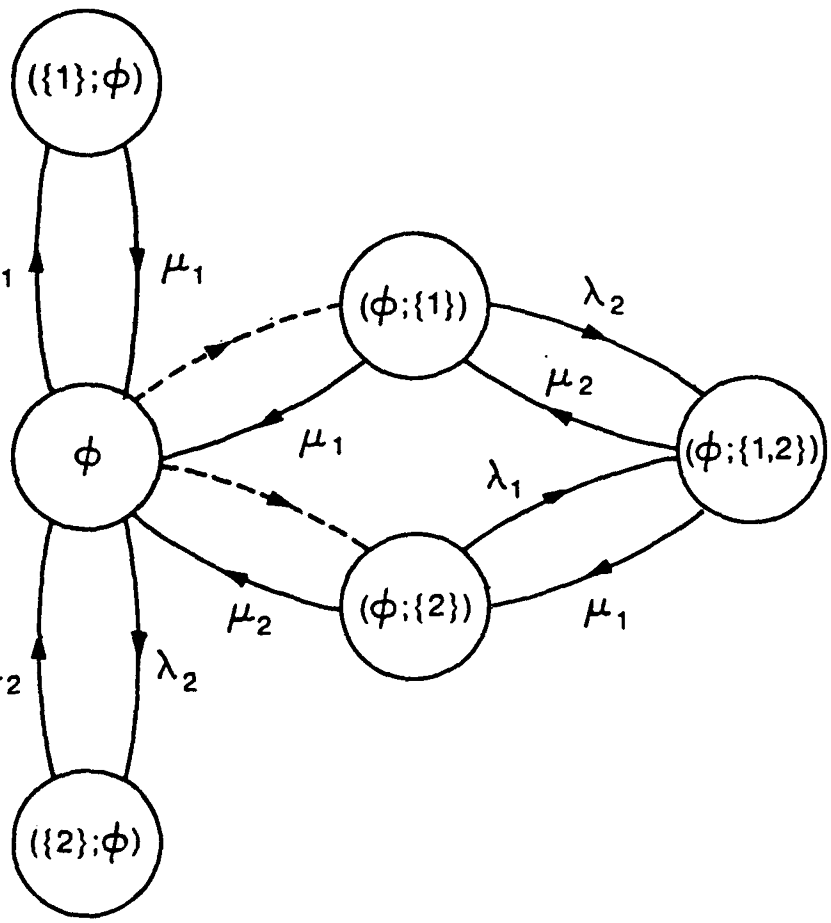


Fig. 6.2 State space for a two node network under disciplined ALOHA and perfect capture

if, in the situation shown, an idle receiver has a nonzero probability of not locking onto an incoming new packet (say, due to the presence of noise), then the dashed

transitions in Figure 6.2 will have nonzero rates, and all the states will be recurrent. If, in a general case, the locking probabilities are such that, for all states $D \in \mathcal{S}$ and links j such that $j \in D$, $j \in L_d(D)$, and j not blocked by D, we have $\mathcal{P}_{d(j)}(D;j) \neq 0$ or $\mathcal{P}_{d(j)}(D;j) \neq 1$, then all transitions from a state $D \in \mathcal{S}$ into the states p-reachable from D have nonzero rates, and all the states in $\mathcal{S}$ are recurrent.

Definition 6.2.8 State $D = (D^+; D^-) \in \mathcal{S}$, with $D^+ = \{l_1, \cdots, l_p\}$ and $D^- = \{l_{p+1}, \cdots, l_q\}$, is said to be *directly p-reachable* if there exists a sequence $D_0 = \phi, D_1, \cdots, D_q = D$ of states in $\mathcal{S}$ and a permutation $(i_1, \cdots, i_q)$ of $(1, \cdots, q)$ such that, for $k = 1, \cdots, q$, we have $D_k = D_{k-1} + l_{i_k}^+$ or $D_k = D_{k-1} + l_{i_k}^-$, according to whether $l_{i_k} \in D^+$ or $l_{i_k} \in D^-$, respectively.

Employing the same arguments used in Section 5.1, we have the following results, whose proof we omit.

Lemma 6.2.9 *If $D \in \mathcal{S}$ is directly p-reachable and $D_1 = (D_1^+; D_1^-)$ is such that $D_1 \subseteq D_2$ (i.e., $D_1^+ \subseteq D^+$ and $D_1^- \subseteq D^-$), then D_1 is directly p-reachable.*

Proposition 6.2.10 *The state space $\mathcal{S}$ consists of ϕ and all states D that are directly p-reachable.*

6.1.3 Balance Equations

In general $\{X(t)\}$ is not irreducible. However, state ϕ can be reached from any state $D \in \mathcal{S}$ in finite time with nonzero probability, by deactivating the links in D. Let $\mathcal{R}$ be the set of states $D \in \mathcal{S}$ that are reachable from ϕ, and $\mathcal{T}$ be the set of those that are not. Any two states in $\mathcal{R}$ communicate, at least through state ϕ, and thus $\mathcal{R}$ is an ergodic class. $\mathcal{T}$ is a transient class, since from each $D \in \mathcal{T}$ there exists a nonzero probability of reaching ϕ, and hence never returning to $\mathcal{T}$. We can then conclude from Corollary 7-9 and Theorem 7-10 of [Heym82] that the

Markov chain $\{X(t)\}$ possesses a unique steady-state distribution. We denote this distribution by $\{p(D) : D \in \mathcal{S}\}$.

Given state $D \in \mathcal{S}$, let $U(D)$ denote the set of links $j \in \mathcal{L}$ such that $j \notin D$ and j is not blocked by D, let $M(D)$ be the set of links $j \in \mathcal{L}$ such that $j \notin D$ and either $D+j^+$ or $D+j^-$ is in the state space $\mathcal{S}$, and let $J(D)$ be the set of links $j \in D$ such that D is p-reachable from $D - \{j\}$. Suppose that $X(t) = D$, and that $j \in U(D)$. Link j will become active in $(t, t+\Delta t)$ with probability $\lambda_j \Delta t + o(\Delta t)$. If $j \in L_d(D)$, the activation of j will lead to state $D + j^+$ with probability $\mathcal{P}_{d(j)}(D; j)$, and will lead to state $D + j^-$ with the complementary probability, independently of the previous history of the system. If $j \notin L_d(D)$, the activation of j will lead into state $D+j^-$ with probability one. If $i \in D$, link i will become inactive in $(t, t+\Delta t)$ with probability $\mu_i \Delta t + o(\Delta t)$, leading the system into state $D - i$. Thus the state at time $t + \Delta t$ is completely determined by the state at time t, and $\{X(t)\}$ is seen to be Markovian. Figure 6.3 shows the transitions out of a generic state D.

We now examine the transitions into state D. We consider separately the transitions from states with one more and one less active link than D.

(i) *States with one more active link:* For each $l \in M(D) \cap L_d(D)$, there is a transition from both $D + l^+$ and $D + l^-$ into D, with rate μ_l. For each link $l \in M(D) \cap \overline{L_d(D)}$, there exists a transition from $D + l^-$ into D, also with rate μ_l.

(ii) *States with one less active link:* Let $j \in J(D) \cap D^+$, which in particular implies that $j \in L_d(D - j)$. Then there exists a transition from state $D - j$ into D, with rate $\mathcal{P}_{d(j)}(D - j; j)\lambda_j$. Let now $j \in J(D) \cap D^-$. If $j \in L_d(D - j)$, there exists a transition from state $D - j$ into D with rate $[1 - \mathcal{P}_{d(j)}(D - j; j)]\lambda_j$; if $j \notin L_d(D - j)$, there exists a transition from state $D - j$ into D with rate λ_j. Additional transitions may exist in the case of undisciplined protocols. In such

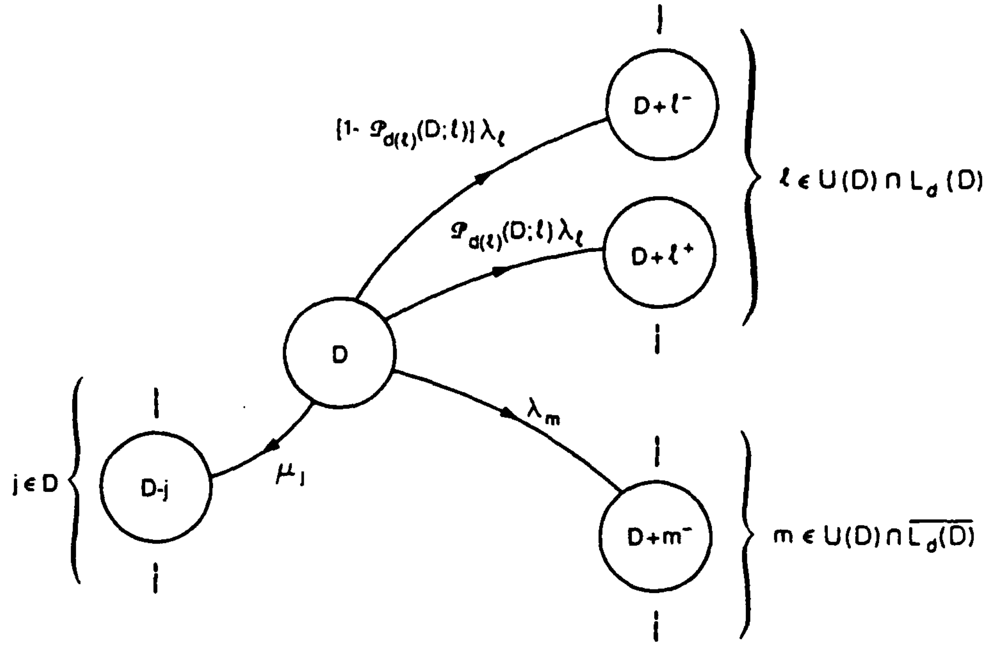


Fig. 6.3 Transitions out of state D

a case if, for any j considered above, there exists $i \in D^- \cap V(s(j))$ and if the state $((D-j)^+ \cup \{i\}; (D-j)^- - \{i\}))$ exists in the state space S and does not block link j, then there exists a transition from this state into state D, occurring with the same rate as the transition from $D-j$ into D.

Figure 6.4 shows the transitions into state D. The solid lines show the transitions that are common to both disciplined and undisciplined protocols. The broken lines represent the transitions that only occur for undisciplined protocols. In the diagram, we designate the states out of which these transitions occur by $D^*(j) - j$.

As an example, we show in Figure 6.5 the state transition rate diagram for a

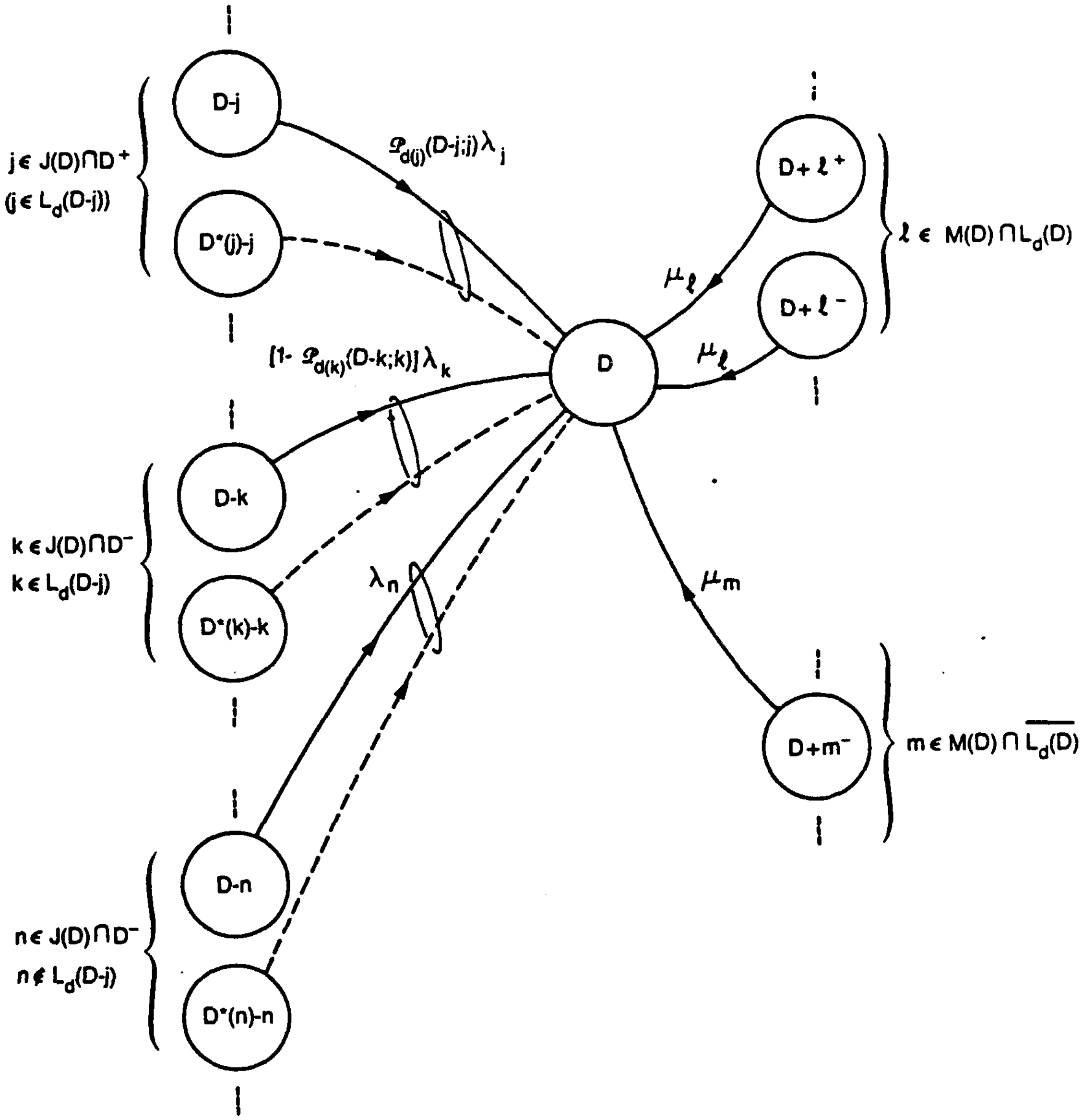


Fig. 6.4 Transitions into state D

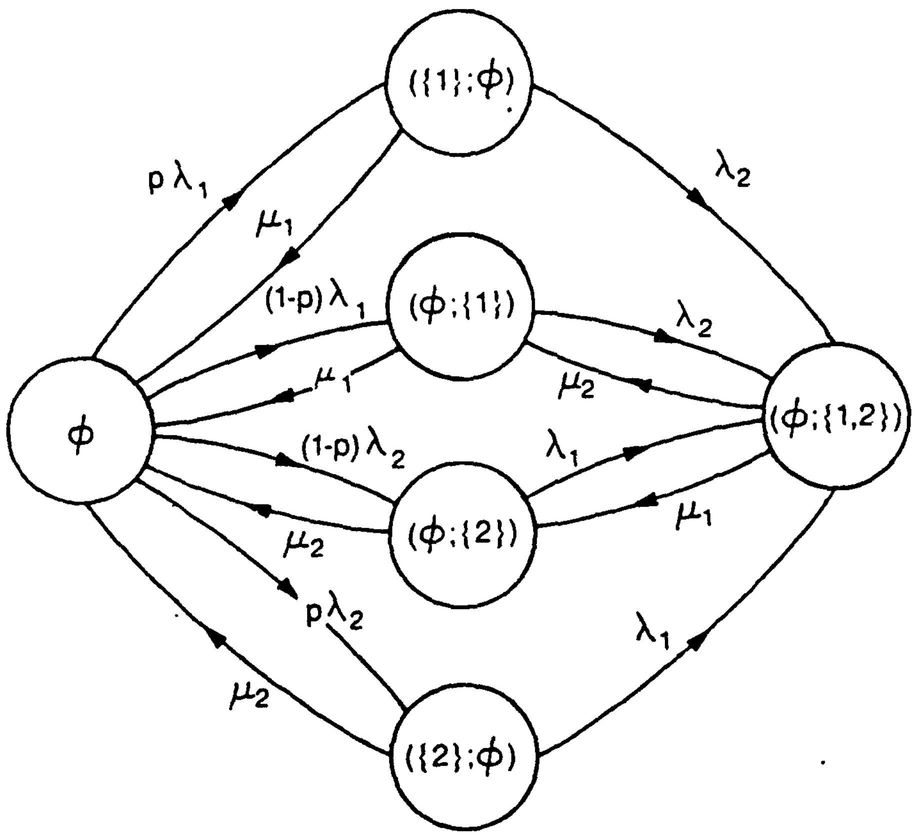


Fig. 6.5 Transition rate diagram for a two node network under ALOHA

two node network under ALOHA in which an idle receiver has a probability p of locking onto a new packet. The transitions referred to above as only existing for undisciplined protocols are those from state $(\{1\};\phi)$ into $(\phi;\{1,2\})$, and from state $(\{2\};\phi)$ into $(\phi;\{1,2\})$.

The balance equations for disciplined protocols, obtained by equating the rate

of transitions into and out of state D, are

$$p(D)\left[\sum_{j\in D}\mu_j+\sum_{i\in U(D)}\lambda_i\right]$$

$$= \sum_{j\in J(D)\cap D^+} p(D-j)\mathcal{P}_{d(j)}(D-j;j)\lambda_j$$

$$+ \sum_{\substack{j\in J(D)\cap D^- \\ j\notin L_d(D-j)}} p(D-j)\lambda_j + \sum_{\substack{j\in J(D)\cap D^- \\ j\in L_d(D-j)}} p(D-j)\left[1-\mathcal{P}_{d(j)}(D-j;j)\right]\lambda_j$$

$$+ \sum_{i\in M(D)} p(D+i^-)\mu_i + \sum_{i\in M(D)\cap L_d(D)} p(D+i^+)\mu_i. \tag{6.1}$$

Due to the cumbersome notation required, we omit the balance equation for undisciplined protocols. The balance equations do not bring any special insight into the problem, and their solutions do not possess special analytical properties. Any study of coupled systems has to resort to a purely algorithmical solution of the problem, which we shall discuss in Chapter 8.

6.1.4 Symmetric Protocols

We consider here a subclass of protocols that presents special advantages from a computational point of view. We shall present here only the their definition and basic properties. The computational advantages of this class of protocols shall be discussed in Section 8.2.1.2.

Definition 6.4.1 We say that a protocol P is *symmetric* if, for any state $D \in \mathcal{L}$ and any $j \in D$, D is p-reachable from $D-j$.

Example 6.4.2 Disciplined ALOHA is a symmetric protocol. Indeed, let $D \in \mathcal{S}$, and let $j \in D$. Since any state $D \in \mathcal{S}$ is admissible, no link $i \in E(s(j)), i \neq j$, is in D, and hence in $D-j$. Similarly, no link $i \in V(s(j))$ is in D^+, and hence in

$(D-j)^+$. But these are the only links that can block link j, and thus link j is not blocked by $D-j$. It is now easy to see that D is p-reachable from $D-\{j\}$, both for the cases where $j \in D^+$ and $j \in D^-$. If $j \in D^+$, then no link $l \in E(d(j))$ is in D, and thus in $D-j$ (i.e., $l \in L_d(D-j)$). Hence D is p-reachable from $D-j$. If $j \in D^-$ then, from Definition 6.2.3, D is p-reachable from $D-j$. Thus Disciplined ALOHA is symmetric.

Example 6.4.3 ID-BTMA, taken as a protocol not in class $\mathcal{D}$, is not symmetric. Let $D \in \mathcal{S}$ and let $j \in D$. We look for all possible links $i \in D, i \neq j$ (and hence $i \in D-j$), such that i blocks j, and thus cause D not to be p-reachable from $D-j$. From the definition of ID-BTMA, the only possible candidates are links $i \in D$ such that (i) $s(i) = s(j)$, (ii) $h_{s(i),s(j)} = 1$, or (iii) $h_{d(i),s(j)} = 1$. Since all states are admissible, condition (i) can never happen. If the hearing matrix is symmetric, any i satisfying (ii) is blocked by j, and thus cannot coexist with j in the same state D, since the one which is activated first will block the other. If the hearing is not symmetric, it is possible for such link i to exist. Condition (iii) can occur if in addition i is such that $h_{d(j),s(i)} = 0$ and j began before i (otherwise i and j cannot coexist in a state D). An example of this situation is obtained from Figure 5.12 by taking i to be link 5 and j to be link 1. Such i and j can coexist in a state D, such as $D = (\{1,5\};\phi)$, where j is blocked by $D-i$. Thus ID-BTMA is not symmetric.

Definition 6.4.4 Given a protocol $\mathbf{P}$, we define the *symmetrization* of $\mathbf{P}$ to be the protocol obtained by redefining the blocking between all pairs of links i and j for which there exists a state $D \in \mathcal{S}$ such that $i \in D$ and $j \in D$, and one of them, say i, blocks j, while j does not block i (otherwise they could not coexist in state $D \in \mathcal{S}$). Here we make the distinction between i and i^+, or j and j^+, if necessary for the purpose of the definition of the blocking between links (see Definition 6.2.2). The symmetrization of $\mathbf{P}$ is defined to be such that any such i and j do not block

each other.

Example 6.4.5 We derive here the symmetrization of ID-BTMA with symmetric hearing. As we saw in Example 6.4.3, given link j and state $D \in \mathcal{S}$, D is not p-reachable from $D - j$ if there exists some $i \in D$ such that $h_{d(i),s(j)} = 1$ and $d(i) \neq d(j)$. We then define the symmetrized protocol to be such that such i does not block j. The definition of the symmetrized protocol in a network with symmetric hearing is thus: link i blocks link j if (i) $s(i) = s(j)$, (ii) $h_{s(i),s(j)} = 1$, or (iii) $h_{d(i),s(j)} = 1$ and $h_{d(j),s(i)} = 1$. This last condition occurs if $d(i) = d(j)$, or in the situation depicted between links 1 and 5 in Figure 5.12.

Example 6.4.6 We consider now the symmetrization of LD-BTMA with symmetric hearing. Let $D \in \mathcal{S}$ and $j \in D$. We again look for links $i \in D$ that block j. From the definition of the blocking in LD-BTMA, we see that any such link i is such that either (i) $i \in D$ and $s(i) = s(j)$, (ii) $i \in D$ and $h_{s(i),s(j)} = 1$, or (iii) $i \in D^+$ and $h_{d(i),s(j)} = 1$. In case (i) link i cannot coexist with j in D, the same happening in case (ii) due to the assumption of symmetric hearing. However, case (iii) is possible and asymmetry in blocking can exist, depending on link j. If $j \in D^+$ then j^+ will also block i if $h_{d(j),s(j)} = 1$. If $j \in D^-$, then j does not block i. Since we want symmetry in the blocking irrespective of whether $j \in D^-$ or $j \in D^+$, we take such i and j as not blocking each other in the symmetrized protocol. Thus the symmetrization of LD-BTMA is defined to be the protocol in which link i blocks link j if (i) $s(i) = s(j)$ or (ii) $h_{s(i),s(j)} = 1$, and is seen to coincide with CSMA.

6.2 Throughput Equations

The derivation of the throughput equations is identical to the one given in Chapter 5. We shall state the main results, omitting the corresponding proofs

when identical to those of Chapter 5.

6.2.1 General Throughput Equations

Let again the throughput S_i of link i be defined as the long-run fraction of time that link i is used for successful transmissions. Given link i, let $\mathcal{U}_s(i)$ be the set of states $D \in \mathcal{S}$ such that i is not blocked by D and no link in $E(d(i))$ is active in D. Thus, for $D \in \mathcal{U}_s(i)$, state $D + i^+$ is p-reachable from D and the corresponding transitions occur with rate $\mathcal{P}_{d(i)}(D;i)\lambda_i$. These are the only transitions of the Markov chain $\{X(t)\}$ that contribute to the throughput S_i.

In order to compute S_i, let us again define $S(D,i)$ as the "partial" throughput given by the long-run fraction of time that link i is used by successful transmissions of i that start with the system in state D, in which case

$$S_i = \sum_{D \in \mathcal{U}_s(i)} S(D,i).$$

In order to compute $S(D,i)$ we consider the regenerative cycles defined by the transitions from D into $D + i^+$. Let $C(D,i)$ be the length of a typical cycle, and $T(D,i)$ the total time during a typical cycle that the channel is used by a successful transmission over link i. Again we have that, with probability one ([Ross70]),

$$S(D,i) = \frac{E[T(D,i)]}{E[C(D,i)]}.$$

Proposition 6.2.1 *The expected cycle length $E[C(D,i)]$ is given by*

$$E[C(D,i)] = \frac{1}{\lambda_i \mathcal{P}_{d(i)}(D;i)p(D)}. \tag{6.2}$$

Proposition 6.2.2 *The throughput S_i of link i is given by*

$$S_i = \lambda_i \sum_{D \in \mathcal{U}_s(i)} p(D)\mathcal{P}_{d(i)}(D;i)E[T(D,i)]. \tag{6.3}$$

As in Chapter 5, the value of $E[T(D,i)]$ is computed from the particular capture model considered.

6.2.2 Spread Spectrum Capture

We consider here the computation of $E[T(D,i)]$ for the capture model of Section 5.3.2.2, whose description we shall not repeat. We compute $E[T(D,i)]$ by means of an absorbing chain $\{X_i^a(t)\}$ with a definition similar to the one in 5.3.2.2.

Let $\mathcal{A}_s(i)$ be the set of states such that $i \in D^+$. The state space of the chain $\{X_i^a(t)\}$ is formed by $\mathcal{A}_s(i)$, together with two absorbing states $\mathcal{A}_c(i)$ and $\mathcal{J}(i)$. $\{X_i^a(t)\}$ enters $\mathcal{A}_c(i)$ whenever a packet loss occurs due to any of the mechanisms described in 5.3.2.2, and enters $\mathcal{J}(i)$ when i's transmission finishes successfully. The mechanisms and rates of the transitions from $D \in \mathcal{A}_s(i)$ into $\mathcal{A}_c(i)$ are: (i) bit errors with rate $\varepsilon(D,i)$, (ii) collisions due to the start of a new transmission over link j, $j \in U(D) \cap \overline{E(d(i))}$, whose timing falls within i's vulnerable window, with rate $p_l(i,j)\lambda_j$, and (iii) packet loss due to the start of a new transmission over link $j, j \in U(D) \cap E(d(i))$, with rate λ_j. The transitions of case (iii) occur only for undisciplined protocols. Transitions into $\mathcal{J}(i)$ occur with rate μ_i from any $D \in \mathcal{A}_s(i)$. Within $\mathcal{A}_s(i), \{X_i^a(t)\}$ transits from $D \in \mathcal{A}_s(i)$ to $D - l$, $l \neq i$, with rate μ_l, to state $D + j^-, j \in U(D) \cap \overline{L_d(D)}$, with rate $[1 - p_l(i,j)]\lambda_j$, and to state $D + j^+, j \in U(D) \cap L_d(D)$ with rate $[1 - \mathcal{P}_{d(j)}(D;j)][1 - p_l(i,j)]\lambda_j$.

$E[T(D,i)]$ is the average time to absorption in $\mathcal{J}(i)$ over the set of sample paths that terminate in $\mathcal{J}(i)$, given that $\{X_i^a(t)\}$ started in $D + i^+ \in \mathcal{A}_s(i)$. Let $\mathbf{R}^s(i)$ be the rate transition matrix of $\{X_i^a(t)\}$. This matrix has the structure shown in Equation (5.25). Similarly to Proposition 5.3.11, we have

Proposition 6.2.3 *Under the capture model of Section 5.3.2.2, the average time the channel is successfully used by i's packet in a cycle, given that the packet is*

successfully locked onto by its destination, is given by

$$E[T(D,i)] = [1 - p_{vp}(D,i)]\overline{\mathbf{T}}_{D+i^+}, \tag{6.4}$$

where $\overline{\mathbf{T}}_{D+i^+}$ *is the component of index* $D + i^+$ *of the column vector*

$$\overline{\mathbf{T}} = \mu_i \, \mathbf{R}_s^{-2}(i) \, \mathbf{1}, \tag{6.5}$$

and $\mathbf{R}_s(i)$ *is the submatrix of the state transition rate matrix of* $\{X_i^a(t)\}$ *whose rows and columns correspond to the states in* $\mathcal{A}_s(i)$.

Results similar to Propositions 5.3.5–5.3.8 can also be derived for the current capture model. Due to their similarity, however, we shall not present them here.

6.3 Summary and Conclusions

We presented in this Chapter the analysis of the more general class of channel access protocols that do not lead to decoupling between the descriptions of the activity of the transmitters and the activity of the receivers. Section 6.1 introduced a Markovian model for the description of the network activity. The state of the corresponding process records the set of active and locked onto links, and the set of active and not locked onto links. The state space of this process was shown to be formed by the states obtained from the idle state by means of link activations only. In this Section we also gave the form of the balance equations for the steady-state probability distribution of this process. In Section 6.2 we presented the derivation of throughput equations, using an approach similar to that of Chapter 5.

Chapter 7

NON-MARKOVIAN SYSTEMS

The previous Chapters studied systems in which the packet lengths and the scheduling delays have exponential distributions, and from which Markovian descriptions result. We examine in this Chapter systems in which one or both of these sets of random variables have general distributions. We only consider the class of protocols studied in Chapter 5, for which the protocol decisions are completely specified by the state of the transmitters. Section 7.2 introduces a class of processes known as Generalized Semi-Markov Processes (GSMPs), that allow the representation of the systems we are interested in. Section 7.3 introduces two constructions for the rescheduling point process, designated as *Continued Renewal Rescheduling* and *Restarted Renewal Rescheduling,* that reduce to a Poisson process for exponential rescheduling intervals, and formulates the resulting transmitter activity processes as GSMPs. Section 7.4 investigates the existence of product form solution for steady-state distribution of these processes, by studying the insensitivity of the corresponding GSMPs with respect to the distributions of the packet lengths and scheduling delays. For *Continuous Renewal Rescheduling* with Poisson rescheduling processes and for *Restarted Renewal Rescheduling* with arbitrary rescheduling

processes these conditions are found to be identical to those found in Chapter 5 for the case of exponential packet lengths. *Continuous Renewal Rescheduling* with arbitrary rescheduling intervals is found not to possess in general a product form solution. Section 7.5 makes a bridge between the processes considered here and some queueing processes, by showing that the process describing the activity of the network links can be obtained as the state of a suitably defined queue.

The original goal of the study of the case of general packet length and scheduling delay distributions was to find throughput expressions that would constitute an extension of those given in Chapter 5 for the exponential case. For this purpose, especially defined GSMPs have to be constructed such that the link throughputs are directly obtained from their steady-state distribution. Unfortunately, these GSMPs are not insensitive with respect to the packet length and scheduling delay distributions, with the consequence that effectively it is not possible in general using this formalism to compute the link throughputs. Nevertheless, we present here the results on the existence of product form solutions due to their intrinsic theoretical interest. In this chapter we use separate notation conventions, to conform to the standard usage in the theory of GSMPs.

7.1 Introduction

Let $X(t)$ be defined, as in Chapter 5, as the set of links which are active at time t. We shall consider in this Chapter the case where link $i \in \mathcal{L}$ has packets whose lengths have a distribution function $B_i(\cdot)$, with

$$\int_0^\infty x\, dB_i(x) = \mu_i^{-1} < \infty,$$

and has scheduling delays whose lengths have a distribution function $A_i(\cdot)$, with

$$\int_0^\infty x\, dA_i(x) = \lambda_i^{-1} < \infty.$$

We shall only require that each $A_i(\cdot)$ and $B_i(\cdot)$ possess a positive density almost everywhere. This condition will ensure the existence and uniqueness of a stationary distribution for $\{X(t) : t \geq 0\}$.

In this general case $\{X(t) : t \geq 0\}$ is no longer Markovian. However, it possesses the structure of a general class of processes known as GSMPs. We shall use general properties of these processes to obtain conditions under which $\{X(t) : t \geq 0\}$ has a product form steady-state distribution. The existence of a product form will be seen to be closely related to the insensitivity of the steady-state distribution with respect to the moments of second and higher orders of the distributions $A_i(\cdot)$ and $B_i(\cdot)$, $i \in \mathcal{L}$.

7.2 Generalized Semi-Markov Processes*

Consider a process $\{X^*(t) : t \geq 0\}$ which, at an arbitrary instant $t \geq 0$, can be in any one of the states g of a finite state space G. Each state g is itself a finite set of elements s of a finite set S. (These elements can represent, for example, links in a packet radio network, or customers in a closed queueing system.) It is required that, for each $s \in S$, there exist at least one $g \in G$ such that $s \in g$. Suppose that $X^*(t) = g$. To each element $s \in g$ (which we shall say to be *active* at time t) there is associated a residual lifetime $Y_s(t) > 0$, determined as described in the following. We let $\mathbf{Y}(t) = (Y_s(t))_{s \in g}$ be the vector of the residual lifetimes of the

*This section is a summary of the main definitions and results in [Scha77] and [Scha78a]. We will try to conform to the notation of these papers whenever possible.

elements active at time t. The lifetimes of the elements which are active at any given time decrease at unit rate†, $X^*(t)$ remaining in state g as long all $Y_s(t)$ are positive. Eventually the lifetime of one of these elements will reach zero (which we will refer to as the "death" of that element), at which time $X^*(t)$ jumps from state g to a new state $g' \in G$. It is assumed that no two elements can die simultaneously. The state transitions are specified by a family of transitions probabilities

$$p = \{p(g, s, \cdot) : g \in G, \ s \in g\},$$

where $p(g, s, g')$ is the probability that the next state of $X^*(t)$ is $g' \in G$ given that the present state is $g \in G$ and that the state transition is caused by the death of $s \in g$. It is required that $p(g, s, g') = 0$ unless $g - \{s\} \subseteq g'$ (i.e., it is required that all other elements which are active in the old state remain active in the new state). Upon entering state g' the residual lifetimes of the elements of g' are determined as follows: the elements in $g - \{s\}$ keep the residual lifetimes they had at the time the state transition took place; a new element $s_i \in g' - (g - \{s\})$ is assigned a residual lifetime which is drawn, independently from the past, from a nonnegative distribution φ_{s_i}, with distribution function $F_{s_i}(\cdot)$, and mean $\eta_{s_i}^{-1} < \infty$. (These residual lifetimes continue to decrease at unit rate after the system enters state g'.) $X(0)$ is chosen to be some arbitrary state $g_0 \in G$, and the initial residual lifetime vector $\mathbf{Y}(0)$ is obtained by assigning to the element $s \in g_0$ a residual lifetime $Y_s(0)$ drawn from the corresponding distribution φ_s. These distributions are assumed to be such that

(C1) no two deaths can occur simultaneously at any time, and

(C2) the resulting $X^*(t)$-process has a unique stationary distribution.

---

†See [Scha78a] for a generalization which allows a countable state space, and arbitrary rates of decrease for the residual lifetimes of the active elements. See also [Hend83] for an alternative construction of a GSMP.

Remark 7.2.1 In general, (C1) is easy to verify directly. A sufficient condition for (C2) is that all distributions involved possess a positive density almost everywhere ([Scha78b]). In some cases it may be possible to obtain less restrictive conditions sufficient to ensure (C2).

Definition 7.2.2 The collection $\Sigma = (G, S, p)$ is called a *generalized semi-Markov scheme (GSMS)*; the process $\{X^*(t) : t \geq 0\}$ is called the *generalized semi-Markov process (GSMP)* based upon Σ by means of the family $\{\varphi_s : s \in S\}$; the process $\{X^*(t), \mathbf{Y}(t) : t \geq 0\}$ is called a *supplemented* GSMP.

Definition 7.2.3 A GSMS is said to be *irreducible* if, for every pair $g, g' \in G$, there exist finite sequences $(g_0, g_1, \ldots, g_n), g_i \in G$, with $g_0 = g$ and $g_n = g'$, and $(s_1, \ldots, s_n), s_i \in S$, such that $p(g, s_1, g_1)\, p(g_1, s_2, g_2) \cdots p(g_{n-1}, s_n, g') > 0$.

Definition 7.2.4 Let $\Sigma = (G, S, p)$ be an irreducible GSMS, and $\Phi(\Sigma)$ the collection of all families $\varphi = \{\varphi_s : s \in S\}$ of distributions concentrated on $(0, \infty)$ which imply the existence of a unique stationary distribution for the corresponding supplemented GSMPs based upon Σ. Let Φ be a nonempty subset of $\Phi(\Sigma)$. Σ is called Φ-*insensitive* if every GSM.˜ based upon Σ by means of an element of Φ has the same stationary distribution.

Notation Let $S' \subseteq S$. We denote by $\Phi_{S'}(\eta_s : s \in S)$ the family of distributions

$$\{\varphi : \varphi \in \Phi(\Sigma), \varphi_s \sim E_{\eta_s}(\cdot) \text{ for } s \notin S', \varphi_s \text{ arbitrary with mean } \eta_s^{-1} \text{ for } s \in S'\},$$

where $E_\eta(\cdot)$ represents an exponential distribution with parameter η. We also set $\Phi_{s_0}(\eta_s : s \in S) \triangleq \Phi_{\{s_0\}}(\eta_s : s \in S)$.

We now state the main results of interest for our applications. It is assumed throughout that the GSMS $\Sigma = (G, S, p)$ is irreducible.

Proposition 7.2.5 *An GSMS Σ is $\Phi_{S'}(\eta_s : s \in S)$-insensitive if and only if Σ is $\Phi_{s'}(\eta_s : s \in S)$-insensitive for every $s' \in S'$.*

Proposition 7.2.6 *An GSMS Σ is $\Phi_{S'}(\eta_s : s \in S)$-insensitive if and only if the stationary distribution of every supplemented GSMP based upon Σ by means of a family $\varphi \in \Phi_{S'}$ is of the form*

$$P\{X(t) = g, Y_s(t) \leq x_s, s \in g \cap S'\} = p_g \prod_{s \in g \cap S'} \eta_s \int_0^{x_s} (1 - F_s(t))\, dt,$$

where $\{p_g : g \in G\}$ is the steady-state probability distribution of the GSMP based upon Σ by means of the exponential family in $\Phi_{S'}(\eta_s : s \in S)$.

Remark 7.2.7 $\{p_g : g \in G\}$ is the normalized solution of the system of equations

$$p_g \sum_{s \in g} \eta_s = \sum_{g' \in G} p_{g'} \sum_{s \in g'} p(g', s, g)\, \eta_s, \quad g \in G. \tag{7.1}$$

These are the global balance equations for the GSMP based upon Σ by means of the exponential family in $\Phi_{S'}(\eta_s : s \in S)$.

Proposition 7.2.8 *Σ is $\Phi_{s_0}(\eta_s : s \in S)$-insensitive if and only if there exists a distribution $\{p_g : g \in G\}$ that satisfies (7.1) and*

$$p_g \eta_{s_0} = \sum_{g' \notin G_0} p_{g'} \sum_{s \in g'} p(g', s, g)\, \eta_s + \sum_{g' \in G_0} p_{g'} p(g', s_0, g)\, \eta_{s_0}, \quad g \in G_0, \tag{7.2}$$

where $G_0 = \{g \in G : s_0 \in g\}$. In the case where such distribution exists, it is the stationary distribution of the GSMP based upon Σ by means of $\Phi_{s_0}(\eta_s : s \in S)$.

Remark 7.2.9 (7.2) is a set of local balance equations for the GSMP based upon Σ by means of the exponential family in $\Phi_{s_0}(\eta_s : s \in S)$, equating the rate of transitions out of state g due to the death of s_0 to the rate of transitions into g due to the birth of s_0.

7.3 Formulation of $\{X(t)\}$ as a GSMP

It is our goal in this Section to formulate a GSMP $\{X^*(t) : t \geq 0\}$ representing the link activity in the network and that, for exponential packet length and scheduling delay distributions, coincides with the process $\{X(t) : t \geq 0\}$ of Chapter 5. The formulation of this GSMP is not unique, and different formulations will possess different properties. We consider two choices, corresponding to different ways of defining the process of the rescheduling points. One such way, that we designate by *Continued Renewal Rescheduling* (CRR), consists of taking the rescheduling point process associated with link i to be a renewal process with interrenewal times possessing the distribution function $A_i(\cdot)$. At each such point the state of the system is examined and, depending on the protocol, a packet transmission either takes place or is inhibited. The fact that the scheduling point process is a renewal process implies that the rescheduling interval after a rescheduling point that gives rise to a packet transmission is drawn from the distribution $A_i(\cdot)$, and thus the next rescheduling point can occur while the packet transmission takes place (in which case it is ignored). An alternative way, which we call *Restarted Renewal Rescheduling* (RRR), does not attempt to generate a new rescheduling point while a packet is being transmitted. Instead, it waits for the end of the packet transmission, at which time, say t, it draws a random interval of length W_r with distribution $A_i(\cdot)$, from which the next rescheduling point is generated, occurring at time $t + W_r$. In this way the rescheduling point process is no longer a renewal process. As we shall see, these two systems differ in some of their properties.

We shall now construct Generalized Semi-Markov Schemes $\Sigma_1 = (G_1, S, p_1)$ and $\Sigma_2 = (G_2, S, p_2)$ such that the associated GSMPs represent the link activity of the CRR and RRR systems, respectively. The states in the collections G_i will be formed by the set of active links, whose residual lifetimes represent the residual

service time of a packet, and by a set of "fictitious" elements, one for each link, whose residual lifetimes keep track of the time left until the next rescheduling point of the corresponding link. In the CRR case, these elements are "alive" in all states, thereby generating a renewal process for the corresponding rescheduling point process. In the RRR case, each such element is only alive as long as the corresponding link is inactive. After a death of the element from which an activation of the corresponding link results, the element is not born again, this event only taking place as the link is deactivated.

7.3.1 Continued Renewal Rescheduling

Let $a_i, i \in \mathcal{L}$, be the element whose deaths correspond to the occurrences of the scheduling points for link i, and generate the state transitions corresponding to the activations of link i if unblocked. Define $\mathcal{A} \triangleq \{a_1, a_1, \cdots, a_L\}$, and let the states $g \in D$ be of the form $g = \mathcal{A} \cup D$, where $D \in \mathcal{S}$ is a set of active links. The death of an element $j \in D$ corresponds to the deactivation of j. The GSMS $\Sigma_1 = (G_1, S, p_1)$ is formally defined in the following way. Let $S \triangleq \mathcal{A} \cup \mathcal{L}$, and $G_1 \triangleq \{g = \mathcal{A} \cup D : D \in \mathcal{S}\}$. The transition probabilities $\{p_1(g, s, g') : g, g' \in G, s \in g\}$ are defined by

$$p_1(g, s, g') = \begin{cases} 1, & \text{if } s = a_j \text{ and } j \in U(D) \text{ and } g' = g \cup \{j\} \\ & \text{or } s = a_j \text{ and } j \notin U(D) \text{ and } g' = g \\ & \text{or } s = j \text{ and } j \in D \text{ and } g' = g - \{j\}, \\ 0, & \text{otherwise.} \end{cases}$$

We associate with this GSMS a family $\varphi = \{\varphi_s : s \in S\}$ of residual lifetime distributions, such that the distribution φ_{a_i}, $i \in \mathcal{L}$, is the scheduling delay distribution $A_i(\cdot)$ for link i and the distribution φ_i, $i \in \mathcal{L}$, is the packet length distribution $B_i(\cdot)$ for link i. Let $\{X^*(t) : t \geq 0\}$ denote the GSMP based upon

Σ_1 by means of the family φ, and suppose that $X^*(t) = \mathcal{A} \cup D$, $D \in \mathcal{S}$. We have that $X^*(t + \Delta t) = \mathcal{A} \cup (D \cup \{i\})$, $i \in U(D)$, if element i became active in the interval $(t, t + \Delta t)$, and $X^*(t + \Delta t) = \mathcal{A} \cup (D - \{j\})$ if element $j \in D$ died in the interval $(t, t + \Delta t)$. The GSMP thus obtained is equivalent to $X(t)$ in the sense that $X^*(t) = \mathcal{A} \cup X(t)$. It follows that, if we again let $\{p(D) : D \in \mathcal{S}\}$ denote the stationary distribution of $\{X(t) : t \geq 0\}$, then $p_{\mathcal{A} \cup D} = p(D)$.

7.3.2 Restarted Renewal Rescheduling

Given $D \in \mathcal{S}$ let $\mathcal{A}_D$ be the set of the elements $a_i \in \mathcal{A}$ such that $i \notin D$. The GSMS $\Sigma_2 = (G_2, S, p_2)$ is defined similarly to Σ_1 of 7.3.1. Let S be as defined in 7.3.1, and $G_2 \triangleq \{g = \mathcal{A}_D \cup D : D \in \mathcal{L}\}$. The transition probabilities $\{p_2(g, s, g') : g, g' \in G, s \in g\}$ are given by

$$p_2(g, s, g') = \begin{cases} 1, & \text{if } s = a_j \text{ and } j \in U(D) \text{ and } g' = g \cup \{j\} - \{a_j\} \\ & \text{or } s = a_j \text{ and } j \notin U(D) \cup D \text{ and } g' = g \\ & \text{or } s = j \text{ and } j \in D \text{ and } g' = g \cup \{a_j\} - \{j\}, \\ 0, & \text{otherwise.} \end{cases}$$

We associate with this GSMS the same family φ of lifetime distributions defined in 7.3.1. The GSMP $\{X^*(t) : t \geq 0\}$ based upon Σ_2 by means of φ is equivalent to $X(t)$ in the sense that $X^*(t) = \mathcal{A}_{X(t)} \cup X(t)$. Their stationary distributions are related by $p_{\mathcal{A}_D \cup D} = p(D)$.

It is immediate to verify that both GSMSs Σ_1 and Σ_2 are irreducible.

7.4 Existence of a Product Form Solution

Equation (5.4) does not make sense in the general case. Here we shall say that $\{X(t) : t \geq 0\}$ has a product form solution if its stationary distribution satisfies

$$p(D) = p(\phi) \prod_{i \in D} \frac{\lambda_i}{\mu_i}, \qquad D \in \mathcal{S}. \tag{7.3}$$

Obviously, the existence of a product form solution for $\{X(t) : t \geq 0\}$ when some set of S' of variables has an arbitrary distribution, all others being exponential, is equivalent to (i) $\Phi_{S'}$-insensitivity of $\{X^*(t) : t \geq 0\}$, and (ii) the existence of a product form solution for the version of the GSMP in which all residual lifetimes are exponential (i.e., the GSMP corresponding to the case where all variables in S' are exponentially distributed, which we shall refer to as the "exponential version" of the GSMP). We shall study separately the cases where (i) only the packet length distributions, and (ii) only the rescheduling delay distributions, are general, the remaining random variables (rescheduling delays in case (i), and packet lengths in case (ii)) having exponential distributions. The following Proposition is valid for both the CRR and the RRR cases.

Proposition 7.4.1 *For arbitrary packet length distributions and exponential scheduling delay distributions, $\{X(t) : t \geq 0\}$ possesses a product form solution if and only if $J(D) = D$, for all $D \in \mathcal{S}$.*

Proof: Let $\{X(t) : t \geq 0\}$ have a product form solution. In particular, the exponential version of the GSMP will have a product form solution. Proposition 5.1.14 then implies that $J(D) = D$, for all $D \in \mathcal{S}$. Conversely, let $J(D) = D$, for all $D \in \mathcal{S}$. Again from Proposition 5.1.14, the exponential version of the GSMP has a product form solution (7.3). We want then to prove that the GSMP is $\Phi_{\mathcal{L}}$-insensitive. Recall that the distribution (7.3) is a solution of the global balance equations (7.1). We

shall now show that (7.3) also satisfies the local balance equations (7.2) for any $i_0 \in \mathcal{L}$ for both the CRR and RRR strategies.

(i) CRR

Let $i_0 \in \mathcal{L}$, and let state $g = \mathcal{A} \cup D$ be such that $i_0 \in D$. The rate of transitions out of g due to the death of i_0 (i.e., due to the termination of i_0's transmission) is $p(D)\mu_{i_o}$. The only states $g' = \mathcal{A} \cup D'$ in which the birth of i_0 will lead into g are those for which $D' = D - \{i_0\}$, in which case a transition into g takes place with rate $1_{\{i_0 \in J(D)\}} \lambda_{i_0}$. Thus the local balance equations (7.2) take the form

$$p_{\mathcal{A} \cup D} \mu_{i_0} = p_{\mathcal{A} \cup (D - (\{i_0\})} 1_{\{i_0 \in J(D)\}} \lambda_{i_0}, \quad D \supset \{i_0\}, \quad D \in \mathcal{S} \tag{7.4}$$

or, given that $J(D) = D$,

$$p_{\mathcal{A} \cup D} = \frac{\lambda_{i_0}}{\mu_{i_0}} p_{\mathcal{A} \cup (D - \{i_0\})}, \quad D \supset \{i_0\}, \quad D \in \mathcal{S} \tag{7.5}$$

which is indeed satisfied by (7.3). Proposition 7.2.8 then allows us to conclude that the GSMP $\{X^*(t) : t \geq 0\}$ is Φ_{i_0}-insensitive for all $i_0 \in \mathcal{L}$ which, together with Proposition 7.2.5, implies that it is $\Phi_{\mathcal{L}}$-insensitive, and hence its stationary distribution is also given by (7.3). Thus $\{X(t) : t \geq 0\}$ possesses a product form solution.

(ii) RRR

Let $i_0 \in \mathcal{L}$, and let state $g = \mathcal{A}_D \cup D$ be such that $i_0 \in D$. Again in this case the only transitions into state g due to the birth of i_0 are from states $g' = \mathcal{A}_{D'} \cup D'$ such that $D' = D - \{i_o\}$. By repeating the argument of (i), we conclude that the local balance equations (7.2) take the form (7.5), and thus that $\{X^*(t) : t \geq 0\}$ is also in this case $\Phi_{\mathcal{L}}$-insensitive. ∎

Proposition 7.4.2 *Continuous Renewal Rescheduling does not possess in general a product form solution for general scheduling delay distributions.*

Proof: We show that insensitivity (and hence a product form solution) does not exist with respect to the distribution of the times between occurrences of two successive scheduling points. For that purpose, let us apply Proposition 7.2.8 by taking s_0 to be a_{i_0}, $i_0 \in \mathcal{L}$. Since the element a_{i_0} is present in all states, the local balance equations (5.2) become

$$p_{\mathcal{A}\cup D}\,\lambda_{i_0} = \sum_{D' \in \mathcal{S}} p_{\mathcal{A}\cup D'}\, p(\mathcal{A}\cup D', a_{i_0}, \mathcal{A}\cup D)\,\lambda_{i_0}, \quad D \in \mathcal{S}.$$

By retaining only the appropriate nonzero probabilities, this equation gives

$$p_{\mathcal{A}\cup D}\,\lambda_{i_0} = \begin{cases} p_{\mathcal{A}\cup D}\,\lambda_{i_0} + p_{\mathcal{A}\cup(D-\{i_0\})}\,\lambda_{i_0}, & \text{if } i_0 \in J(D) \\ 0, & \text{if } i_0 \in U(D) \\ p_{\mathcal{A}\cup D}\,\lambda_{i_0}, & \text{if } i_0 \in \mathcal{L} - U(D) - J(D) \end{cases} \tag{7.5}$$

Clearly this system of equations is not compatible with (7.1), since for D such that $i_0 \in U(D)$, (7.5) requires $p_{\mathcal{A}\cup D} = 0$, whereas (7.1) is known to have a strictly positive solution. The conclusion then follows from Proposition 7.2.8. ∎

Proposition 7.4.3 *Restarted Renewal Rescheduling with general rescheduling delay distribution and exponential packet length distributions possesses a product form solution if and only if $M(D) = U(D)$, for all $D \in \mathcal{S}$.*

Proof: Let $\{X(t) : t \geq 0\}$ have a product form solution. By Proposition 5.1.14, $M(D) = U(D)$ for all $D \in \mathcal{S}$. Conversely, let $M(D) = U(D)$ for all $D \in \mathcal{S}$. Again from Proposition 5.1.14 we have that the exponential version of the GSMP has a product form solution. We now show that the GSMP is $\Phi_{\mathcal{A}}$-insensitive. Let $i_0 \in \mathcal{L}$, and consider a_{i_0}. Let $g = A_D \cup D$ be such that $a_{i_0} \in g$, which in particular implies

$i_0 \notin D$. For the construction of the local balance equations (7.2) we have to look for the states $g' = \mathcal{A}_{D'} \cup D'$ that can lead into g via the birth of a_{i_0}. This event can happen in one of two ways: (i) in the transition from $g' = \mathcal{A}_{D\cup\{i_0\}} \cup (D \cup \{i_0\})$ into g, if $i_0 \in M(D)$, and (ii) in the transition from $g' = g$ into g, if $i_o \notin U(D)$. Thus the local balance equations become (recall that $U(D) \subseteq M(D)$)

$$p_{\mathcal{A}_D \cup D}\, \lambda_{i_0} = \begin{cases} p_{(\mathcal{A}_D - \{a_{i_0}\})\cup(D\cup\{i_0\})}\, \mu_{i_0}, & \text{if } j \in U(D), \\ p_{(\mathcal{A}_D - \{a_{i_0}\})\cup(D\cup\{i_0\})}\, \mu_{i_0} + p_{\mathcal{A}_D \cup D}\, \lambda_{i_0}, & \text{if } j \in M(D) - U(D), \\ p_{\mathcal{A}_D \cup D}\, \lambda_{i_0}, & \text{if } j \notin M(D). \end{cases} \tag{7.6}$$

Since by hypothesis $M(D) = U(D)$, (7.6) reduces to

$$p_{(\mathcal{A}_D - \{a_{i_0}\})\cup(D\cup\{i_0\})} = \frac{\lambda_{i_0}}{\mu_{i_0}}\, p_{\mathcal{A}_D \cup D}, \quad i_0 \in U(D)$$

and

$$0 = 0, \quad i_0 \notin M(D)$$

which is compatible with the solution (7.3) of the global balance equations (7.1). ∎

As a direct consequence of Propositions 5.1.13, 7.4.1, and 7.4.2 we have (note that only the blocking properties of the protocol, and not the form of the service time distributions, are relevant for the proof of Proposition 5.1.13)

Proposition 7.4.4 (Criterion for the existence of a product form—Continuous Renewal Rescheduling) *Under Continuous Renewal Rescheduling in a system with general packet length distributions and exponential scheduling delay distributions, a necessary and sufficient condition for a channel access protocol, together with a given network topology and traffic requirements to have a product form solution is that, for all pairs of used links i and j, link j blocks link i whenever link i*

blocks link j. In general, a product form solution does not exist when some of the scheduling delay distributions is not exponential.

From Propositions 5.1.13, 7.4.1, and 7.4.3 we obtain

Proposition 7.4.5 (Criterion for the existence of a product form—Restarted Renewal Rescheduling) *Under Restarted Renewal Rescheduling in a system with general packet length and scheduling delay distributions, a necessary and sufficient condition for a channel access protocol, together with a given network topology and traffic requirements, to have a product form solution is that, for all pairs of used links i and j, link j blocks link i whenever link i blocks link j.*

Remark 7.4.5 It is possible for a given network configuration and access protocol to be insensitive with respect to the packet length distributions of a proper subset of the links of the network, and nevertheless not have a product form solution. In terms of (7.1) and (7.2), this corresponds to the solution of (7.1) satisfying (7.2) for some, but not all, links of the network. As an example, consider the network of Figure 5.1 operating under the ID-BTMA protocol, and with nonzero traffic requirements over all links. It is easy to see that, when i_0 is taken to be either link 3 or 4, the corresponding system (7.4) is compatible with the solution (7.3) of (7.1), and hence insensitivity exists with respect to the packet length distributions of these links. On the other hand, if we take i_0 to be any of the other links in the network, it is always possible to find states $D \in S$ such that $i_0 \notin J(D)$, for which (7.4) then requires $p(D) = 0$. This requirement is incompatible with (7.3), thus showing that insensitivity does not exist with respect to links 1, 2, 5 and 6.

Remark 7.4.6 In the construction of a GSMP given in Section 7.2, an element of $s \in S$ is assigned, at the times of its birth, lifetimes that can be viewed as the interarrival times in the renewal point process associated with the distribution φ_s. It can be shown that the results presented in Section 7.2 remain valid if the

successive lifetimes assigned to an element $s \in S$ are obtained as the interarrival times of an *arbitrary* stationary point process with intensity η_s ([Scha78b]). This implies, in particular, that an insensitive system, as defined in Section 7.2, is also insensitive with respect to the choice of the stationary point process from which the successive lifetimes of a given active element are generated.

7.5 Formulation of $\{X(t)\}$ as a Queueing Process

The process $\{X(t) : t \geq 0\}$, can be formulated as the queueing process of an infinite server queue with state-dependent arrivals. Although this formulation does not yield new results over the one presented in Section 7.3 and 7.4, it establishes the connection between the process of the active links and some processes studied in the queueing literature. Consider an infinite server queue with L classes of customers (recall that L represents the number of used links in the network), each class being uniquely associated with each used link in the network, and vice-versa. The successive service times of the customers of class i are i.i.d. random variables with distribution function $B_i(\cdot)$. The arrival processes depend on the state of the queue, and differ according to whether a CRR or RRR strategy is used. The state of the queue is defined as the set of classes of customers present in the queue.

Under CRR, each customer class has associated with it a renewal process in which the interrenewal times have the distribution of the scheduling delay of the link corresponding to that customer class. A renewal point for the process associated with link i gives rise to a customer arrival if and only if at that renewal epoch the state D of the queue does not block link i.

Under RRR, arrivals of class i are generated as follows: as soon as a class i departure occurs, a renewal process is started with interrenewal times with distri-

bution $A_i(\cdot)$. The first such renewal time that finds the queue in a state D in which i is not blocked results in a class i customer arrival, at which time the renewal process is stopped. When the customer departs, the process is repeated.

When the scheduling delay distributions are exponential, both of these methods give rise to nonhomogeneous Poisson arrival processes, whose rate at time t is a function of the queue occupancy at that time, in the following manner: if D is the set of customer classes present in the queue at time t, then the arrival process for class i customers has rate λ_i whenever the corresponding set of links D (in the packet radio network) does not block link i, and has rate 0 otherwise.

If we denote by $X(t)$ the set of customer classes present in the queue at time t, and restrict $X(0)$ to belong to the collection $\mathcal{S}$ (defined in Section 5.1.1), then at any subsequent time t we still will have $X(t) \in \mathcal{S}$. It is easily seen that this process $\{X(t) : t \geq 0\}$ coincides with the process defined in 7.1 in terms of the link activity of the packet radio network. The correspondence between both models is made by interpreting the arrival of a class i customer to the queue as the activation of link i in the packet radio network, and its departure as the deactivation of the same link. This formulation explains some of the similarities between the properties of the process describing the joint activity of the transmitters in the network, and the properties of some queueing systems, e.g., those considered in ([Chan77]). ([Kell76]) also studies queues with state-dependent arrivals which are equivalent to the ones described here in the cases where a product form solution exists.

7.6 Summary and Conclusions

We studied in this Chapter the existence of a product form solution for the steady state probabilities of the transmitter activity process for protocols of class

$\mathcal{D}$, when the packet lengths or the rescheduling intervals do not possess an exponential distribution. The resulting non-Markovian processes belong to a class known as Generalized Semi-Markov Processes (GSMP's). We gave in Section 7.2 the definition of a GSMP, and presented the properties of a GSMP that are of interest for our applications. The main such result states that the steady-state distribution of a GSMP depends only on the means of the random variables that define it if and only if the (Markov) process obtained by replacing those random variables by exponential variables with the same means satisfies a given form of local balance. In Section 7.3 we introduced two constructions for the rescheduling point process, designated as *Continued Renewal Rescheduling* and *Restarted Renewal Rescheduling,* that reduce to a Poisson process for exponential rescheduling intervals, and formulated the resulting transmitter activity processes as GSMPs. We then applied in Section 7.4 the insensitivity property stated above to the study of the conditions under which each of the rescheduling mechanisms considered leads to a product form solution. For *Continuous Renewal Rescheduling* with Poisson rescheduling processes and for *Restarted Renewal Rescheduling* with arbitrary rescheduling processes these conditions were found to be identical to those found in Chapter 5 for the case of exponential packet lengths, i.e., symmetry in link blocking. *Continuous Renewal Rescheduling* with arbitrary rescheduling intervals was found not to possess in general a product form solution. In Section 7.5 we formulated the transmitter activity process as the state of a queue with state-dependent arrivals, thus establishing a connection between the processes considered in this work and some processes considered in the queueing literature.

Chapter 8

NUMERICAL COMPUTATION: ALGORITHMS

The previous Chapters focused on the derivation of expressions for the link throughputs, given the network specification and the sets $\{\mu_{ij}^{-1}\}_{(i,j)\in\mathcal{L}}$ of average message lengths and $\{\lambda_{ij}\}_{(i,j)\in\mathcal{L}}$ of link rescheduling rates. We discuss in this Chapter algorithms for the numerical evaluation of these expressions and determination of network capacity. Most of the algorithms presented in this Chapter have been incorporated into a general computer package for the capacity analysis of packet radio networks.

Section 8.2 is devoted to the computation of the link throughputs given the channel access protocol, the capture mode, and the operating parameters (rescheduling rates and average message lengths) of the network. Subsection 8.2.1 considers the more general protocols of Chapter 6, which do not lead to decoupling between the activity of the transmitters and the activity of the receivers. These protocols require the numerical setting up and solution of the balance equations of the processes involved. The enumeration of the state space of these processes is reduced to the traversal of a directed graph, and a breadth-first search (bfs) is considered for this

purpose. A number of properties of the bfs enumeration of the state space are then established. From these properties, a characterization of the order in which states are enumerated and an efficient algorithm for that enumeration are derived. Finally, algorithms and data structures appropriate for a computer-implemented enumeration of the state space, setting up of the balance equations, solution of the systems of linear systems involved, and computation of throughput, are described. Subsection 8.2.2 considers the restricted class of protocols considered in Chapter 5, which lead to decoupling between the activity of the transmitters and of the receivers. Its contents parallel those of Subsection 8.2.1. Section 8.3 considers the problem of, given a set of link throughput requirements, solving for the network operating parameters that attain those requirements. This Section presents a fixed-point iteration algorithm for the solution of the problem. Section 8.4 considers the problem of finding the capacity corresponding to an *a priori* given traffic pattern. Two strategies are presented: (*i*) a trial-and-error binary search method, at each step of which the feasibility of a tentative value of the capacity is tested, and (ii) a parametric method that, given an arbitrary linear functional h of the rescheduling rates, finds the network throughput as a function of the value assigned to h, and then obtains the capacity by maximization over that value.

8.1 Introduction

We consider algorithms for the solution of the following three basic types of problems, each one of which builds upon the previous ones for its solution:

1. **Link throughput computation:** given a set of rescheduling rates, determine the resulting link throughputs.

2. **Solution for desired throughputs:** given a set $\{S_{ij}\}_{(i,j)\in\mathcal{L}}$ of desired link throughputs, *determine* if it is feasible; if feasible, find the set of link rescheduling rates that attains it.

3. **Capacity determination:** given a traffic pattern matrix $\mathbf{A} = [\alpha_{ij}]$ compatible with the hearing matrix $\mathbf{H} = [h_{ij}]$, find the supremum of the set of values of S for which the set of link throughputs $\{S_{ij} = S\,\alpha_{ij} : (i,j) \in \mathcal{L}\}$ is feasible.

In 3. above, the matrix $\mathbf{A}$ is said to be *compatible* with $\mathbf{H}$ if $h_{ij} = 0$ implies $\alpha_{ij} = 0$.

The solution of 1. requires the solution of the global balance equations of the associated Markov chain, and the computation of the average successful channel utilization per transmission attempt. For the purpose of the latter, we consider throughout this Chapter the capture model of Section 5.3.2.2. We shall discuss mainly the more general case of protocols not in class $\mathcal{D}$ and capture modes not in class $\mathcal{C}_f$, for which we shall give algorithms and data structures for the construction of the state space and the Q-matrix of the corresponding Markov chains, the setting up of the systems of equations (6.5), and the evaluation of (6.3). We shall then discuss briefly the solution of 2., viewed as a system of nonlinear equations

$$S_{ij} = G_{ij}\,f_{ij}(\mathbf{G}), \quad (i,j) \in \mathcal{L}$$

on the unknowns $\mathbf{G}$, with $\{S_{ij}\}_{(i,j)\in\mathcal{L}}$ given. Finally, we shall present two methods for the solution of 3. and discuss their relative merits.

8.2 Link Throughput Computation

8.2.1 General Protocols

The general case of protocols not in class $\mathcal{D}$ requires the solution of the global balance equations (6.1) for the steady-state probabilities, the solution of the system of equations (6.5) for the average successful durations of the different link transmissions, and the computation of the sums in (6.3). As implemented in a computer program, the computations comprise mainly two stages, the details of which we examine in the following sections:

1. **Setup:** construction of the state space and Q-matrix of the main and auxiliary Markov chains;

2. **Computation:** solution of the sets of linear equations giving the steady-state probabilities and the packet average successful lengths, and computation of (6.3).

8.2.1.1 The Enumeration of the State Space as a Graph-Theoretical Problem

Consider the directed graph (digraph for short) G corresponding to the state transition rate diagram of the Markov chain under consideration. This digraph has a set of vertices $V = \mathcal{S}$, and a set of edges E such that the arc (D, D') from D into D', with $D, D' \in \mathcal{S}$, is in E if and only if the transition from D into D' is allowed under the rules of the access protocol*. A traversal of the digraph G will provide an *enumeration* of the state space $\mathcal{S}$, with different traversals giving different orderings of the states, and thus different structures for the resulting transition

*Note that, under some capture modes, some of these transitions may have zero probability of occurrence.

rate matrix (or Q-matrix). At the same time, the successive examination of the arcs emanating from each vertex found during the traversal process will give an enumeration of the nondiagonal entries of the Q-matrix. We can thus reduce the problem of construction of the state space and the Q-matrix to that of traversal of the vertices and labeling of the arcs of the digraph G.

The two most important types of systematic graph traversals are depth-first and breadth-first [Aho83]. For reasons to be discussed later, we shall adopt for our algorithms a breadth-first traversal, or breadth-first search (*bfs*). Given its importance, we give now a brief description of it.

At any stage of a breadth-first search, each vertex in G is marked either *visited* or *unvisited,* and either *searched* or *unsearched.* Given a *visited* and *unsearched* vertex v, the algorithm marks that vertex as *searched* and visits, in succession, all vertices adjacent from* v which are still *unvisited,* marking them as *visited.* When all vertices adjacent to v have been exhausted, a previously *visited* but *unsearched* vertex is selected, and the process is repeated. The *visited unsearched* vertices are selected to be searched in the order in which they were first visited. If at some stage no more *visited unsearched* vertices exist, an existing *unvisited* vertex is selected as starting vertex and marked *visited,* and the procedure is repeated until no more *unvisited* vertices are left. Initially all vertices are marked *unvisited* and *unsearched.* Given a starting vertex, the algorithm visits it (marking it as *visited*), and then proceeds to apply to it the procedure described above.

As an example, consider the diagraph of Figure 8.1. Let the unvisited vertices adjacent to a vertex be visited in alphabetical order of their labels, and vertex A be chosen as the starting vertex. Vertex A will then be marked *visited* and *searched,* and vertices B and E visited (and so marked), in this order. Since there are no

*Given two vertices i and j of a directed graph such that there is an arc from i into j, vertex i is said to be *adjacent to* vertex j, and vertex j is said to be *adjacent from* vertex i (cf. [Law176]).

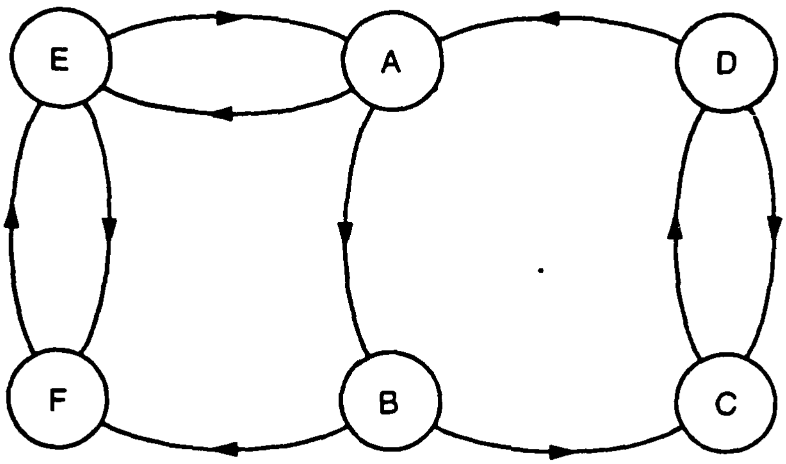


Fig. 8.1 Digraph for example of breadth-first search

more unvisited vertices adjacent from A, the next *visited* and *unsearched* vertex, in this case B, is marked *searched*, and all unvisited neighbors of it are visited. In this way, vertices C and F are marked *visited*, and no more *unvisited* vertices adjacent from B exist. The next *visited* and*unsearched* vertex is E, which is thus marked *searched*, but which does not have any *unvisited* vertices adjacent from it. The next *visited unsearched* vertex is C, which is then marked *searched*, and from which D is visited, and thus marked. Vertices F and C, still marked *unsearched*, are next searched, but no more unvisited vertices exist. Thus the breadth first traversal yields the enumeration $ABECFD$ of the vertices of the digraph. Note that, as the vertices adjacent to each vertex being searched are examined to determine whether they have been visited or not, the edges of the graph are also enumerated.

For a question of clarity, we shall consider in the following the construction of the state space S separately from the construction of the Q-matrix, although in practice the two are done simultaneously.

8.2.1.2 Enumeration of the State Space

Given the vertex being currently searched, we are required by the *bfs* algorithm

to visit (and thus generate) all unvisited vertices adjacent to it. In our case, the vertices adjacent to a given vertex (state) $D \in \mathcal{S}$ are obtained by either deactivating a link $i \in D$, or activating a link $j \in U(D)$. The following Lemma tells us that only the states obtained by activating a link $j \in U(D)$ need to be considered when looking for unvisited states adjacent to state D, as long as the breadth-first search is started in state ϕ. In the following, N designates the number of nodes in the network.

Lemma 8.2.1 *Let $c(D)$ be the number of links active in state D, and let $\mathcal{S}_m \triangleq \{D \in \mathcal{S} : c(D) = m\}$, for $m = 0, \ldots, N$. When performing a breadth-first search of the state space, started in state ϕ, the following are true:*

(i) Any state in $\mathcal{S}_n, n = 1, \ldots, N$, is visited (and thus searched) before any other state in $\mathcal{S}_m, m > n$, and after any other state in $\mathcal{S}_k, k < n$;

(ii) When any state in $\mathcal{S}_n, n = 1, \ldots, N$, is searched, only states in $\mathcal{S}_{n+1}$ are visited from it (i.e., all states in $\mathcal{S}_m, m < n$, have been visited);

(iii) When all the states in $\mathcal{S}_n, n = 1, \ldots, N$, have been searched, all states in $\mathcal{S}_{n+1}$ have been visited.

Proof: (iii) is a direct consequence of the fact that, by Proposition 6.2.10, any state in $\mathcal{S}_n$ is adjacent from (i.e., p-reachable from) some state in $\mathcal{S}_{n-1}$. Rather than give a formal proof of (i) and (ii) by induction, it is easier to verify them by following the steps that the algorithm takes during the search, keeping in mind that (i) the set of all vertices which are adjacent to the vertices of $\mathcal{S}_k$ is $\mathcal{S}_{k-1} \bigcup \mathcal{S}_{k+1}$, and (ii) the states are searched in the order that they are visited:

0: Visit ϕ;

1: Search states in $\mathcal{S}_0 = \{\phi\}$. During this search, all states in $\mathcal{S}_1$ are visited;

2: Search states in $\mathcal{S}_1$. During this search, (i) all states in $\mathcal{S}_2$ are visited, and (ii) only states in $\mathcal{S}_2$ are visited, since all of $\mathcal{S}_0 \bigcup \mathcal{S}_1$ had been visited before;

$\vdots$

k: Search states in $\mathcal{S}_{k-1}$. During this search, (i) all states in $\mathcal{S}_k$ are visited, and (ii) only states in $\mathcal{S}_k$ are visited, since all of $\mathcal{S}_0 \bigcup \cdots \bigcup \mathcal{S}_{k-1}$ had been visited before;

$\vdots$

∎

In principle, in order to know if a state D', obtained from state $D \in \mathcal{S}$ by activating some link $j \in U(D)$, has already been visited (i.e., generated), we should have to search explicitly the set of already existing states for state D'. The repetition of this search for every such pair of states D and D' (of the same cardinality) can be computationally very expensive. The following Lemmas state that, for symmetrical protocols and under the conditions stated, this search is not necessary.

Lemma 8.2.2 *Assume an (arbitrary) ordering of the network links. For all $k \geq 0$, when searching a vertex $D \in \mathcal{S}_k$, let the vertices in $\mathcal{S}_{k+1}$ adjacent to D be enumerated (i.e., visited) in the order that results from the successive activation of the links in $U(D)$ in the order of increasing link numbering. In the cases in which the activation of a link can lead to two states, one with the link not locked onto, and the other with the link locked onto, let the former be enumerated before the latter. Let also $l_1(D)$ be the lowest-numbered, $l_2(D)$ the second lowest-numbered, etc., link active in state D. The following statement is valid for symmetric protocols and for a breadth-first search started in state ϕ. Let D_1 and D_2 possess the same number of active links, and let k be the smallest index such that either (i) $l_k(D_1) \neq l_k(D_2)$,*

or (ii) $l_k(D_1) = l_k(D_2) = j$*, with* j *being locked onto in one of the states, and not locked onto in the other state. In case (i), if* $l_k(D_1) < l_k(D_2)$ *then* D_1 *is visited before* D_2*. In case (ii), if* $j \in D_1^-$ *and* $j \in D_2^+$ *then* D_1 *is visited before* D_2*.*

Proof: The proof is by induction on the common number of links active in D_1 and D_2. The conclusion is true for states in $\mathcal{S}_1$ since, by construction, given two links i and j, with $i < j$, (i) any of the states $(\{i\};\phi)$ or $(\phi;\{i\})$ is visited before any of the states $(\{j\};\phi)$ or $(\phi;\{j\})$, and (ii) state $(\phi;\{i\})$ is visited before state $(\{i\};\phi)$. Assume the conclusion true for states containing n active links, and let D_1 and D_2 contain $n+1$. By the definition of breadth-first search, each of D_1 and D_2 is visited from the lowest-numbered state that they are adjacent from. We look now for those states. Since we are dealing with symmetric protocols, any $D \in \mathcal{S}$ can be reached from the state obtained by deactivating any of the links $j \in D$. Furthermore, for undisciplined protocols, if there exists $i \in D^- \cap V(s(j))$ then a transition may exist from state $((D-j)^+ \cup \{i\};(D-j)^- - \{i\})$ into state D. These states need not however be considered when looking for the lowest-numbered state that D is adjacent from, since the application of the induction hypothesis shows that state $D-j$ has a number lower than any of these states. Thus we only need to consider the states $D-j$, for $j \in D$. Let D be a generic state in $\mathcal{S}_{n+1}$, with $i,j \in D$, and $i<j$. By the application of case (i) of the induction hypothesis to $D-i$ and $D-j$, we see that $D-i$ is visited *after* $D-j$. Thus the lowest-numbered state that D is adjacent from is obtained by deactivating the *highest-numbered* link in D. We now apply this result to both D_1 and D_2, and denote by $a(D_1)$ and $a(D_2)$ the states they are visited from. If k is such that $l_k(D_1)$ and $l_k(D_2)$ are the highest-numbered links in both D_1 and D_2, then $a(D_1) = a(D_2)$. The order in which links are activated when generating the vertices adjacent from a given vertex implies the validity of (i) and (ii). If $l_k(D_1)$ and $l_k(D_2)$ are not the highest-numbered links in D_1 and D_2, then the application of the induction hypothesis to $a(D_1)$ and $a(D_2)$

implies the validity of (i) and (ii). ∎

Corollary 8.2.3 *Let $h(D)$ be the highest-numbered link active in state D. Under the assumptions of Lemma 8.2.2 and when searching state D, all unvisited states adjacent to state D are obtained by activating all the links $j \in U(D)$, with $j > h(D)$.*

Proof: As argued in the proof of the previous Lemma, given a state D, its "ancestor" $a(D)$ is obtained by deactivating the highest-numbered link in D. This means that from any state D', only states obtained by activating some link $j > h(D')$ are visited when searching D'. ∎

Corollary 8.2.4 *Consider state $D \in S$, and let links $j, l \in U(D)$, be such that $j < l$. For a disciplined protocol, state $D + j$ (meaning either $D + j^+$ or $D + j^-$) is visited before state $D + l$ (meaning either $D + l^+$ or $D + l^-$).*

Proof: Let k be such that, in the notation of Lemma 8.2.2, $j = l_k(D+j)$. Then we have that $l_i(D+j) = l_i(D+l)$ for $i = 1 \cdots, k-1$, and that $l_k(D+j) = j < l_k(D+l)$. Thus, from Lemma 8.2.2, state $D + j$ is visited before state $D + l$. ∎

Note that Corollary 8.2.4 does not hold for undisciplined protocols. As an example, consider the four node ring of Figure 5.12 under ALOHA. Let $D = (\{2\}; \phi)$, and consider the activation of links 6 and 8. The activation of link 6 leads to states $(\{2,6\}; \phi)$ and $(\{2\}; \{6\})$. The activation of link 8 leads to states $(\{8\}; \{2\})$ and $(\phi; \{2,8\})$. Even though $8 > 6$ we have, by Lemma 8.2.2, that state $(\{8\}; \{2\})$ is visited *before* state $(\{2\}; \{6\})$.

From the preceding Lemmas we derive Algorithm 8.2.5, shown in Figure 8.2, for the enumeration of the state space for symmetric protocols. During the execution of this algorithm, the visited states are stored in an array S in the order they are visited. The variables *current* and *last* represent, respectively the indices in this array of the state being currently searched, and of the last state generated

Algorithm 8.2.5

```
begin;
    number the links, from 1 to L;
    current := 1; last := 1; S[1] := φ;
    repeat
        D := S[current];
        for i := 1 to L do begin
            if (i ∈ U(D)) and (i > h(D)) then begin
                last := last + 1;
                D' := D + i⁺;
                S[last] := D';
                if i ∈ L_d(D) then begin
                    last := last + 1;
                    D' := D + i⁺;
                    S[last] := D';
                end;
            end;
        end;
        current := current + 1;
    until (current > last);
end.
```

Fig. 8.2 Algorithm for the enumeration of the state space for symmetric protocols

(visited), and $h(D)$ represents the number of the highest-numbered link active in D. Algorithm 8.2.5 relies on the assumption that each state $D \in S$ is adjacent (i.e., p-reachable) from any other state obtained by deactivating any link $j \in D$. The algorithm will thus not perform correctly *as is* for nonsymmetric protocols. Suppose, however, that for a given nonsymmetric protocol and network topology it is possible to order the links of the network in such a way that, whenever link j blocks link i but i does not block j, link j has a number higher than link i. It is easy to see that for this case the algorithm will perform correctly. One way to verify the existence of such ordering makes use of the *nonsymmetrical interference graph* of the network links, defined as the directed graph with vertices corresponding

to the links in the network, and where there is an edge from link i to link j if j blocks i but i does not block j. An ordering of the links as the one desired is possible if and only if the graph has no cycles, in which case it represents a partial ordering of the network links between which asymmetric blocking exists. The existence of cycles can be checked easily by performing a depth-first search on the nonsymmetrical interference graph [Aho83]. In the general case, however, such an ordering is not possible, and an algorithm for the enumeration of the state space will have the drawback of having to check, for each state D being searched and each link $j \in U(D)$, whether the state(s) D' resulting from the activation of j already exist. Furthermore, such algorithm will not provide in general a criterion, such as the one presented in Lemmas 8.2.1 and 8.2.2, to determine which of two given states is visited first. The construction of the state space for nonsymmetric protocols can be reduced, however, to the one for symmetric protocols by using Algorithm 8.2.4 applied to the symmetrization of the protocol under consideration. The transition rates between states in the state space of the symmetrized protocol are governed by the blocking properties of the original protocol. This is the approach followed in our implementation of the analysis. The resulting state space will contain in general some additional states not contained in the state space of the protocol originally considered, but these states will be transient, and their existence will not alter the result of any relevant computations.

8.2.1.3 Construction of the State Transition Rate Matrix

As noted in Section 8.2.1.1, the enumeration of the nondiagonal entries of the Q-matrix is equivalent to the enumeration of the arcs of the digraph G corresponding to the state transition rate diagram. This enumeration can be done systematically while performing a traversal (using any suitable algorithm) of the vertices of the

Algorithm 8.2.6

```
begin
    D := D_0;      { initial state for traversal }
    repeat
        for each i ∈ D do begin
            D' := D − i;
            Q[D, D'] := μ_i;
        end;
        for each i ∈ U(D) do begin
            D' := D + i^-;
            if (i ∉ L_d(D)) then begin
                Q[D, D'] := λ_i;
            end else begin
                Q[D, D'] := (1 − P_{d(i)}(D; i))λ_i;
                D' := D + i^+;
                Q[D, D'] := P_{d(i)}(D; i)λ_i;
            end;
        end;
        if (all vertices visited) then begin
            stopcondition := true;
        end else begin
            D := next(D);
        end;
    until (stopcondition);
end.
```

Fig. 8.3 Algorithm for the enumeration of the entries of the Q-matrix for symmetric protocols

digraph G, and then enumerating the arcs going out of each visited vertex. Assume that, given state D, the function $next(D)$ gives the next state to be visited during some traversal of the state space. Algorithm 8.2.6, shown in Figure 8.3, gives a schematic description of the enumeration of the entries of the Q-matrix for symmetric protocols. This algorithm, when used with the breadth-first search of 8.2.1.2, produces a Q-matrix with a tridiagonal block structure in which the diagonal blocks D_k, $k = 1, \ldots, N$, are diagonal matrices, with all the μ_i's in the blocks

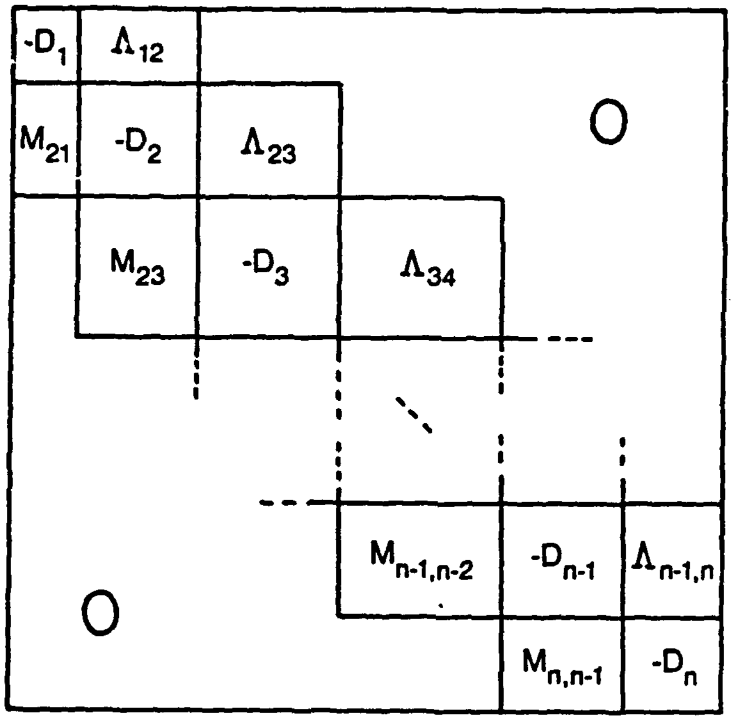


Fig. 8.4 Block structure of Q-matrix produced by breadth-first search

$M_{k-1,k}$, $k = 2, \ldots, N$, to the left of the main diagonal, and all the λ_j's in the blocks $\Lambda_{k,k+1}$, $k = 1, \ldots, N-1$, to the right of the main diagonal (Figure 8.4). This structure presents favorable numerical properties ([Cont72]), and does not exist in the case when the state space is enumerated using a depth-first search.

8.2.1.4 Data Structures and Algorithms

We now describe the PASCAL data structures and algorithms used in a computer implementation of the analysis of general symmetric protocols.

State space and Q-matrix: The state space is stored in an array *StateArray*, which is also used for accessing the entries of the Q-matrix, as described below. The cell *StateArray*[i] is a record, containing one field in which the i-th state in

the state space (according to the order in which states are enumerated) is stored, and two pointer fields, one pointing to the corresponding row of the Q-matrix, and the other pointing to the corresponding column. Due to the special structure of the problem, it is possible to devise a number of storage strategies for the Q-matrix that offer significant advantages over a straightforward approach. A first step consists of noting that the Q-matrix is sparse. Indeed, the size of the state space grows typically as $e^{\alpha L}$, where L is the number of network links and α is some positive constant. However, each row has at most $2L$ nonzero nondiagonal entries, and so the fraction of nonzero entries is of the order of Le^{-2L}, and becomes arbitrarily close to zero as L increases. Thus a substantial savings on the storage requirements can be obtained by storing only the nonzero entries of Q. Some additional simplification results if we note that the nondiagonal entries of the Q-matrix are of the form $\alpha\lambda_i$ or μ_i, $i = 1, \cdots, L$. The value of α depends on the transition in question and can either be $\alpha = 1$ or, for some state $D \in \mathcal{S}$, $\alpha = \mathcal{P}_{d(i)}(D, i)$ or $\alpha = 1 - \mathcal{P}_{d(i)}(D, i)$. Thus in a Q-matrix cell we can just store the factor α, together with the index i, if the corresponding rate is λ_i, or just the index $-i$, if the corresponding rate is μ_i. The actual rates λ_i and μ_i are stored in an array *RateArray*, such that $RateArray[i] = \lambda_i$, $i = 1, \cdots, L$, and $RateArray[-i] = \mu_i$, $i = 1, \cdots, L$. In this way an updating of the entries of the Q-matrix is not necessary when new reschedulng rates or average packet lengths are desired, while at the same time the updating of the entries of the array *RateArray* is very simple. Some further reduction in the number of entries stored can be accomplished by noting that, for the symmetric protocols considered here, a transition from D to $D-j$ implies the reverse transition, and that, for disciplined protocols, a transition from D into $D+j^+$ or $D+j^-$ implies the reverse transition. Thus, for disciplined protocols, one single entry can represent both $Q[D, D']$ and $Q[D', D]$ and only one-half of the matrix has to be stored. This strategy can be used for undisciplined protocols by adding an additional flag to each

entry to indicate the case where, for some $D \in S$ and $j \in U(D)$, only the transition from D into $D+j^+$ or $D+j^-$ exists, without the corresponding reverse transition. Such a situation occurs if and only if some link $i \in V(s(j))$ is active in state D, and is thus very easily recognizable.

The cells (entries) of the Q-matrix are stored as a doubly linked list, linked by both columns and rows. This arrangement is required by the fact that the traversal of a row (or column) in the full Q-matrix corresponds to the traversal of both a row *and* column in the half-matrix that we store. The first cell in a given row or column is accessed by means of the pointers in the cell of the corresponding state in the array *StateArray*. Each cell of the Q-matrix is also a record, with the following fields: (i) the index of the row to which the cell belongs, (ii) the index of the columns to which the cell belongs, (iii) the index of the link to which the entry refers (iv) the coefficient, $P_{d(i)}(D;i)$ or $1-P_{d(i)}(D;i)$, that affects the corresponding λ-entry, (v) a flag that for undisciplined protocols is **true** if the cell represents only one transition, as described above, and **false** otherwise, (vi) a pointer to the next cell in the same column, and (vii) a pointer to the next cell in the same row.

Stationary probability vector: The solution of the balance equations (6.1) is stored in an array *ProbArray*, in which *ProbArray*[i] is the steady state probability of the state stored in *StateArray*[i].

Average successful length vector: During the computation of the throughput of link i, the solution of (6.5) is stored in an array *TArray* in which, if we let i be the index in *StateArray* of state D, *TArray*[i] stores the value of $\overline{T}_D$, as defined in Proposition 6.2.3. This array is reused when computing the throughputs of different links.

Q-diagonal vector: The values of the diagonal entries of the Q-matrix are stored in an array *QdiagArray*, in which *QdiagArray*[i] stores the negative sum of the

transition rates out of the state stored in *StateArray*[i], *in the Markov chain* $\{X(t)\}$. (In the chain $\{X_i^s(t)\}$ there are additional transitions out of state $D \in A_s(i)$, with rate $\epsilon(D; i)$, due to bit errors.) This array is recomputed each time a new set of rescheduling rates is introduced, by traversing the rows of the (full) Q-matrix and summing the rates corresponding to the existing cells.

Unblocked list: For the computation of the sums in Equation (6.3) we need to know the indices, in *StateVector*, of the states $D \in \mathcal{U}_s(i)$ and $D + i^+ \in \mathcal{A}_s(i)$. In order to avoid an expensive search in *StateArray* each time one such index is needed, we keep an array *Unblocklist* of linked lists, one for each link, in which *Unblocklist*[i] contains the information necessary for the computation of (6.3) for link i. Each cell in the list for link i corresponds to a given $D \in \mathcal{U}_s(i)$, and is a record with the following fields: (i) the index, in *StateArray*, of state D, (ii) the index, in *StateArray*, of $D + i^+$, (iii) the value of $\mathcal{P}_{d(i)}(D; i)$, and (iv) a pointer to the next cell. Whenever a cell is inserted in the Q-matrix representing the activation of link i in the locked state, the indices of the origin and destination state are recorded in a cell which is inserted in the list *Unblocklist*[i]. This procedure involves no extra computational effort, as is apparent from Algorithm 8.2.6. These lists are also used for the implicit generation of the matrices $\mathbf{R}_s(i)$ of (6.5) from the Q-matrix, as discussed later.

Generation of state space, Q-matrix, and Unblocked list: All these three data structures are simultaneously generated. Algorithm 8.2.7, shown in Figure 8.5, is used for this purpose. This Algorithm is obtained from a combination of Algorithms 8.2.5 and 8.2.6. In it, we let $i(D)$ designate the index of state D in the array *StateArray*, *Qcell*[i, j] denote the (i, j)-th entry of the Q-matrix, and *Ucell* denote a generic cell of *Unblocklist*. All other symbols have the same meanings as in Algorithms 8.2.5 and 8.2.6. The ordering of the states afforded by the generation

Algorithm 8.2.7

```
begin;
    number the links, from 1 to L;
    current := 1; last := 1; S[1] := φ;
    repeat
        D := StateArray[current];       { state being currently searched }
        for j := 1 to L do begin
            if (j ∈ U(D)) then begin
                D' := D + j⁻;
                if (j < h(D)) or (V(j) ∩ D⁺ ≠ φ) then begin
                                        { D' already exists in state space }
                    index := i(D');     { find position of D' in StateArray }
                end else begin;     { add new state to state space }
                    last := last + 1;
                    StateArray[last] := D';
                    index := last;
                end;
                create cell[current, index]; insert into Q-matrix;
                if (j ∈ L_d(D)) then begin
                    last := last + 1;       { add transition with j locked onto }
                    D' := D + j⁺;
                    StateArray[last] := D';
                    if (j < h(D)) then begin      { D' already exists in state space }
                        index := i(D');      { find position of D' in StateArray }
                    end else begin;      { add new state to state space }
                        last := last + 1;
                        StateArray[last] := D';
                        index := last;
                    end;
                    create Qcell[current, index]; insert into Q-matrix;
                    add Ucell to Unblocklist;
                end;
            end;
        end;
        current := current + 1;
    until (current > last).
    end.
```

Fig. 8.5 Algorithm for the enumeration of the state space and construction of Q-matrix for symmetric protocols

of the state space via a breadth-first search, and which is expressed in Lemmas 8.2.1 and 8.2.2, presents advantages that are apparent during the execution of Algorithm 8.2.7. In particular, the linear ordering of the states in *StateArray* allows the determination of $i(D)$ to be made via a binary search, with a computational complexity which is logarithimic in the number if states involved. Furthermore, since all states with k elements occur as a block in *StateArray* before all states with $k+1$ elements and after all those with $k-1$ elements, the search for $i(D)$ may be confined to the block where D is known to belong to. Another simplification results for disciplined protocols from Corollary 8.2.4. When filling out the row of the Q-matrix corresponding to state $D \in S$ by activating the links $i \in U(D)$ in order of ascending number, the resulting states $D + j$ also come out in ascending order, in such a way that the new cell to be inserted in the Q-matrix can just be appended to the end of the list representing the row. In the case of undisciplined protocols the resulting states may not appear in increasing order, and the new cell may have to be inserted in the middle of the list representing the row. However, from the proof of Corollary 8.2.4 it is easy to see that this situation only occurs when the link j being activated is such that there exists $i \in D^+ \cap V(s(j))$, and is thus easily recognizable.

Solution of the auxiliary Markov chain $\{X_i^a(t)\}$: When solving the system of linear equations (6.5) it is advantageous to keep the matrices involved as sparse as possible. Thus it is not appropriate to solve (6.5) as

$$\mathbf{T} = (\mathbf{R}_s^2(i))^{-1}\mathbf{1},$$

since the matrix $\mathbf{R}_s^2(i)$ will lose sparsity with respect to $\mathbf{R}_s(i)$ and the computation of $\mathbf{R}_s^2(i)$ is expensive. Instead, the solution of (6.5) can be obtained from the successive solution of two systems of linear equations, as in

$$\mathbf{T} = \mathbf{R}_s^{-1}(i)[\mathbf{R}_s^{-1}(i)\,\mathbf{1}]. \tag{8.1}$$

For iterative methods of solution of $\mathbf{R}_s^{-1}\,\mathbf{1}$ that only involve performing the products between a vector and a matrix, the matrix $\mathbf{R}_s(i)$ does not have to be explicitly generated and stored. As described in Proposition 6.2.3, the set of rows/columns of $\mathbf{R}_s(i)$ is formed by the rows/columns of the Q-matrix that correspond to the states in $\mathcal{A}_s(i)$. The entries of $\mathbf{R}_s(i)$ below the main diagonal are identical to the corresponding entries of the Q-matrix, whereas the entries above the main diagonal, representing transitions from $D \in \mathcal{A}_s(i)$ into $D + j \in \mathcal{A}_s(i)$, $j \in U(D)$, are related to the entries of the Q-matrix by

$$\mathbf{R}_s(i)[D, D+j] = [1 - p_l(i,j)]\, Q[D, D+j]. \tag{8.2}$$

Let $\mathbf{v} = (v_D)_{D \in \mathcal{A}_s(i)}$ be a vector for which we want to compute the product $\mathbf{R}_s(i)\,\mathbf{v}$. The vector $\mathbf{v}$ is stored in an array *workvec* with the dimension of the state space $\mathcal{S}$, such that if i is the index of D in *StateArray*, the element v_D is stored in *workvec*[i]. The product $\mathbf{R}_s(i)\,\mathbf{v}$ is computed by multiplying the Q-matrix by the vector *workvec*, but working only with the components where indices correspond to the states in $\mathcal{A}_s(i)$. These indices are found in a straightforward way by a traversal of the linked list *Unblocklist.* When multiplying an entry of $\mathbf{R}_s(i)$ by an entry of $\mathbf{v}$, one only has to retrieve the corresponding entry of the Q-matrix and the factor $[1 - p_l(i,j)]$ of Equation (8.1), if it applies. The latter can be stored in a table, for example. The diagonal entries of $\mathbf{R}_s(i)$ are the sum of the corresponding entry of the Q-matrix and the negative bit error rate $\varepsilon(D, i)$, and also do not have to be explicitly stored.

Throughput computation: Algorithm 8.2.8, shown in Figure 8.6, gives the sequence of computations for the evaluation of the link throughputs for a given set of values of the rescheduling rates and average packet lengths, in a situation of nonperfect capture. It assumes all data structures to have been properly initialized. Once the steady state probabilities and the average successful channel utilization

Algorithm 8.2.8

begin
  solve steady state equations; store in *ProbArray*;
  for $i := 1$ **to** L **do begin**
    solve $\mathbf{T} = \mathbf{R}_s^{-2}\,\mathbf{1}$; store in *TArray*;
    compute S_i via Equation (6.3);
  end;
end.

Fig. 8.6 Sequence of computations for evaluation of link throughput

per transmission are computed, the computation of S_i is made by traversing the linked list *Unblocklist*[i] and, for each cell in the list, retrieving the corresponding elements from the arrays *ProbArray* and *TArray*, as well as the value of $P_{d(i)}(D, i)$ stored in the cell of *Unblocklist*. Under perfect capture only the solution of the balance equations is required, since in this case $\overline{T}(D, i) = \mu_i^{-1}$.

8.2.2 Protocols of Class $\mathcal{D}$

The throughput of protocols of class $\mathcal{D}$ can be obtained by treating them as general protocols not in $\mathcal{D}$, and using a solver for the latter. This was the approach taken in our computer implementation of the capacity analysis of packet radio networks. However, substantial reductions in the cost of the computation can result if the special structure of the protocols in $\mathcal{D}$ is exploited.

Many of the aspects of the computation of the throughput for protocols of class $\mathcal{D}$ are similar to those encountered for the protocols not in $\mathcal{D}$. We discuss in this section mainly the aspects peculiar to protocols in $\mathcal{D}$. For these protocols, the link throughputs are given, in general, by Equations (5.35) or (5.36). The computation of the link throughput may require the solution of either or both of

the systems of equations (5.1) and (5.37), for the steady-state probabilities and the average successful channel utilizations per transmission *attempt, respectively.* The solution of (5.1), necessary in situations where a product form solution does not exist, or the solution of the auxiliary Markov chain (5.37), necessary in situations where the probability of success of a packet depends on its length, require the enumeration of the state space and construction of the Q-matrix of the process $\{X(t)\}$ of Section 5.1. The solution of (5.37), neccessary for protocols not in class $\mathcal{C}_s$, requires the enumeration of the state space and construction of the Q-matrix of the processes $\{Y_n(t)\}$, $n = 1, \cdots, N$, of Section 5.2.

8.2.2.1 Enumeration of State Space and Construction of Q-matrix of $\{X(t)\}$

The issues in the enumeration of the state space and construction of the Q-matrix for the process $\{X(t)\}$ of Chapter 5 are similar to those encountered for the state representation of Chapter 6, for protocols not in class $\mathcal{D}$. In particular, there exists a subclass of protocols, corresponding to the symmetric protocols of Chapter 6, for which the enumeration of the state space is more easily done. This subclass is formed by the protocols that lead to a product form solution. Algorithms 8.2.4–8.2.6 and the associated data structures,applicable to symmetric protocols, can be translated directly into algorithms and data structures applicable to the product form protocols by changing appropriately the nature of the states, and by disregarding the distinction between the activation of a link in the locked and the unlocked state. As an example, we show in Figure 8.7 an algorithm for the simultaneous construction of the state space and Q-matrix for product form protocols, directly *obtained from Algorithm* 8.2.7. The discussion of Section 8.2..2 regarding the enumeration of the state space for nonsymmetric protocols also applies directly to the case of protocols that do not lead to a product form solution. In particular it is

Algorithm 8.2.9

```
begin
    number links, from 1 to L;
    current := 1; last := 1; StateArray[1] := φ;
    repeat
        D := StateArray[current];
        for j := 1 to L do begin
            if (j ∈ U(D)) then begin
                D' := D ∪ {j};
                if (j < h(D)) then begin
                    index := i(D');
                end else begin
                    last := last + 1;
                    StateArray[last] := D';
                    index := last;
                end;
                create cell[current, index]; insert into Q-matrix;
            end;
        end;
        current := current + 1;
    until (current > last);
end.
```

Fig. 8.7 Algorithm for the enumeration of state space and construction of Q-matrix for product form protocols

again true that, if it is possible to find an ordering of the links such that whenever link i blocks link j and link j does not block link i then $j < i$, Algorithm 8.2.9 will perform correctly. However, such ordering does not exist in general. An algorithm for protocols with nonsymmetric blocking is computationally expensive since, whenever considering a state obtained from an existing state via the activation of some link one does not know in general whether the resulting state already exists in the state space, and thus a search among the existing states is required. Thus in such a situation it is preferable to use Algorithm 8.2.9 applied to the symmetrization of the protocol in question, even at the expense of possibly introducing some transient

states in the state space.

8.2.2.2 Evaluation of Product Form Solution

Equation (5.6) can be used for the computation of the steady state probabilities whenever a product form solution exists. The evaluation of (5.6) requires the computation of the normalizing factor $p(\phi)^{-1}$, given by Equation (5.7). If the enumeration of the state space is required by some other computations, then $p(\phi)^{-1}$ can be directly evaluated via (5.7). Otherwise, one can avoid that enumeration by the use of the following recursive algorithm, which is a direct adaptation to the state description employed here of an algorithm first given in [Boor80]. Let V be a set of network links, and define

$$SP(V) \triangleq \sum_{\substack{D \in \mathcal{S} \\ D \subseteq V}} \prod_{i \in D} \frac{\lambda_i}{\mu_i}, \tag{8.2}$$

with $SP(\phi) = 1$. Let also $N(i)$ be the set of all links that block link i. It is easy to verify that

$$SP(V) = SP(V - \{i\}) + \frac{\lambda_i}{\mu_i} SP(V - N(i)). \tag{8.3}$$

Let $D \in \mathcal{S}$ be a state for which we want to compute $p(D)$. If we let $\mathcal{V}$ denote the set of all network links, then

$$p(\phi)^{-1} = SP(\mathcal{V})$$

and

$$p(D) = \frac{1}{SP(\mathcal{V})} \prod_{i \in D} \frac{\lambda_i}{\mu_i}.$$

[Boor83] presents an algorithm and data structures for the efficient computation of (8.2) based on (8.3). [Kers84] also introduces a recursive relation that constitutes a generalization of (8.3).

8.2.2.3 Steady State Probabilities of $\{Y_n(t)\}$

Once the steady state probabilities of $\{X(t)\}$ are computed, the steady state probabilities $\{p_n(D;j) : D \in \mathcal{S}, j \in D \cap V(n)\}$ of each $\{Y_n(t)\}, n = 1, \cdots, N$, can be computed from Equations (5.13) or (5.15). The latter is obtained from the former by a simple rescaling of the unknowns, and is thus equivalent to it. If a product form solution exists for $\{p(D) : D \in \mathcal{S}\}$ then Equation (5.16) can be used directly instead of (5.13) or (5.15).

The solution of (5.13), (5.15), or (5.16), for any two different nodes can be done independently, and in parallel. Data structures similar to those of Section 8.2.1 can be used for the storage of the state space and the coefficient matrices of these systems. Some differences, however, exist. One such difference is that, unlike the balance equations encountered thus far, the coefficient matrices *are not* transition rate matrices of Markov processes, and the corresponding systems of equations are not homogeneous. Another difference is that the matrices of the coefficients do not have the symmetric structure of the matrices considered in 8.2.1, and thus the use of separate cells for the storage of the elements above and below the main diagonal is required. In order to obtain an ordering of the states with properties similar to those of Section 8.2.1.2, the underlying state space $\mathcal{S}$ should be the state space of the "symmetrization" of the protocol in use, where the symmetrization $\mathbf{P}_{sym}$ of a protocol $\mathbf{P}$ in $\mathcal{D}$ is defined to be the protocol in which given links $i, j \in \mathcal{L}$, these links block each other in $\mathbf{P}_{sym}$ if they block each other in $\mathbf{P}$, and do not block each other in $\mathbf{P}_{sym}$, otherwise. The generation of the matrices of coefficients of (5.13), (5.15), or (5.16), for symmetric protocols can be made using algorithms similar to those of Section 8.2.1, and shall not be discussed here.

As a side remark, it should be noted that if we apply the formalism of Chapter 6 to a protocol in $\mathcal{D}$, the size of the resulting state space is much larger than the sum

of the sizes of the state spaces $\mathcal{S}, \mathcal{S}_1, \cdots, \mathcal{S}_N$ of Chapter 5. Indeed, let links $l_1, \cdots, l_k$ have different destinations, and let D be such that $l_1, \cdots, l_k$ are active in state $\mathcal{D}$ and no link emanating from their destination node is active. In the state space $\mathcal{S}$ of the chain $\{X(t)\}$ of Chapter 6 there will be 2^k states with the same active links as state D, corresponding to the different combinations in which each link $l_1, \cdots l_k$ can be locked onto or not. In the state space of the processes $\{Y_n(t)\}$ of Chapter 5, there will be only $2k$ such states, namely $(D; \phi) \in \mathcal{S}_{d(l_i)}$, and $(D; l_i) \in \mathcal{S}_{d(l_i)}$.

8.2.2.4 Evaluation of Throughput Equations

In the cases where the solution of (5.1), (5.13) or (5.37) is required, data structures and algorithms similar to those of Section 8.2.1 can be used, which we shall not discuss. In a situation where none of these are required, such as the one of a protocol with a product form solution under idealistic perfect capture, the computations can be greatly simplified. Suppose that, for each link $i \in \mathcal{L}$, the throughput S_i is given by

$$S_i = G_i \sum_{D \in \mathcal{U}_s(i)} p(D),$$

with $G_i \triangleq \frac{\lambda_i}{\mu_i}$, and there exists a collection of links $\mathcal{L}(i)$ such that the set $\mathcal{U}_s(i)$ can be represented as $\mathcal{U}_s(i) = \{D \in \mathcal{S} : D \subseteq \mathcal{L}(i)\}$. The throughput equations then become

$$S_i = G_i \sum_{\substack{D \in \mathcal{S} \\ D \subseteq \mathcal{L}(i)}} p(\phi) \prod_{j \in D} G_j$$

or, in the notation of 8.2.2.2,

$$S_i = G_i \frac{SP(\mathcal{L}(i))}{SP(\mathcal{V})}, \tag{8.4}$$

where $\mathcal{V}$ denotes the set of all network links. The numerator and denominator of the fraction in (8.4) can then be easily computed using the product form recursion (8.3) or any refinements of it, such as the ones described in [Boor83].

The sets of states $\mathcal{U}_s(i)$ and the sets of links $\mathcal{L}(i)$ involved in the computation of (8.4) depend on the protocol considered. We saw in Section 5.3.1.2 that, for idealistic perfect capture with symmetric hearing, $\mathcal{U}_s(i)$ is formed by all the states $D \in \mathcal{S}$ that do not contain any links in $\mathcal{C}(i)$, where $\mathcal{C}(i)$ *is the set of links* j *such* that either (i) j blocks link i, or (ii) the source node of j can be heard by i's destination. For CSMA, $\mathcal{C}(i)$ is formed by all links whose source node is within one hop of i's source or i's destination. For C-BTMA, $\mathcal{C}(i)$ is the set of all links whose source nodes are within two hops of i's source. For these protocols the throughput equations can be written as

$$S_i = G_i \frac{SP(\mathcal{V} - \mathcal{C}(i))}{SP(\mathcal{V})}, \quad i \in \mathcal{L}.$$

8.3 Solution for Desired Throughputs

Let $\mathbf{S} = [S_i]_{i \in \mathcal{L}}$ be a given vector of desired link throughputs. We want to *find out* whether it is feasible and, if so, what rescheduling rates allow the given throughputs to be attained. This problem can be formulated as determining the feasibility and, if feasible, the solution of the system of nonlinear equations

$$S_i = G_i \, f_i(\mathbf{G}), \quad i \in \mathcal{L}, \tag{8.5}$$

where $\mathbf{G} = [G_i]_{i \in \mathcal{L}}$ is the vector of normalized rescheduling rates $G_i \triangleq \frac{\lambda_i}{\mu_i}$, and the functions $f_i(\cdot)$ are derived from (5.35) or (6.3). These functions can be computed using algorithms as the ones given in Section 8.2. We consider the quantities

Algorithm 8.3.1

begin
  $k := 0$;
  $\mathbf{G}^{(0)} := \mathbf{S}$;
  $S_i^{(0)} := G_i^{(0)} f_i(\mathbf{G}^{(0)}, \qquad i \in \mathcal{L}$;
  repeat
    $k := k + 1$;
    $G_i^{(k)} := G_i^{(k-1)} S_i / S_i^{(k-1)}$;
    $S_i^{(k)} := G_i^{(k)} f_i(\mathbf{G}^{(k)}), \qquad i \in \mathcal{L}$;
  until $(\|\mathbf{S}^{(k)} - \mathbf{S}\| < \epsilon \, \|\mathbf{S}\|)$ **or** $(divcondition(\mathbf{G}^{(k)}, \mathbf{G}^{(k-1)}, \mathbf{G}^{(k-2)}, \mathbf{G}^{(k-3)}))$;
end.

Fig. 8.8 Fixed-point iteration algorithm for solution for desired throughputs

$\{\mu_i^{-1}\}_{i \in \mathcal{L}}$ to be given *a priori*, and do not explicitly represent the dependence of the link throughputs on them.

The system of Equation (8.5) can be rewritten as

$$G_i = \frac{S_i}{f_i(\mathbf{G})}, \quad i \in \mathcal{L} \tag{8.6}$$

and solved using fixed-point iteration ([Cont72]), as in the algorithm of Figure 8.8. In this Algorithm, $\mathbf{G}^{(k)}$ represents the rescheduling rates computed in the k-th iteration, and $\mathbf{S}^{(k)}$ represents the link throughputs corresponding to $\mathbf{G}^{(k)}$. The iteration process is stopped when the norm of the difference between the vectors of the desired and the obtained link throughputs is less than some prespecified tolerance ϵ, or when the divergence of the algorithm is detected as specified by the function *divcondition*. This function becomes **true** when (i) for some link $i \in \mathcal{L}$ the rescheduling rate $G_i^{(k)}$ exceeds some large number $MaxG$ fixed *a priori*, (ii) a prespecified maximum number of iterations is exceeded, or (iii) the ratios $G_i^{(k+1)}/G_i^{(k)}$ increase for three consecutive values of k, for all $i \in \mathcal{L}$. Condition (iii) was arrived

at empirically, and is found to work very well in practice. A heuristic motivation for this condition will be given in the next Section. It should be noted that the determination of the divergence condition does not require the storage of the three last sets of rescheduling rates obtained from the iteration process, but only the storage of two boolean flags that record whether such increases occurred in the two previous iteration steps.

When (8.6) converges, it converges necessarily to a solution of (8.5). We do not know of any proof of the converse statement, namely that if (8.5) has a solution, then (8.6) converges. Nevertheless, this appeared to be the case in all the situations we encountered.

8.4 Capacity Determination

We assume now a traffic pattern vector $\mathbf{A} = [\alpha_i]_{i \in \mathcal{L}}$ to be given. We want to find the capacity C corresponding to the traffic pattern $\mathbf{A}$. This problem can be formulated as that of finding

$$C \triangleq \sup_{\mathbf{G} \geq \mathbf{0}} \{ S(\mathbf{G}) : S = G_1 f_1(\mathbf{G})/\alpha_1 \} \tag{8.7}$$

subject to the restrictions

$$\frac{G_1 f_1(\mathbf{G})}{\alpha_1} = \frac{G_2 f_2(\mathbf{G})}{\alpha_1} = \cdots = \frac{G_L f_L(\mathbf{G})}{\alpha_L}. \tag{8.8}$$

Alternatively, it can be formulated as finding the supremum of the values of S such that the system of nonlinear equations

$$\alpha_i S = G_i f_i(\mathbf{G}), \quad i \in \mathcal{L}, \tag{8.9}$$

Algorithm 8.4.1

```
begin
    low := 0;
    high := 1.1/ max_{n∈{1,...,N}} Σ_{i∈E(n)} α_i;
    mid := (low + high)/2;
    repeat
        S := mid · A;
        perform Algorithm 8.2.7 with input S;
        if (divcondition) then begin
            high := mid;
        end else begin
            low := mid;
        end;
        mid := (low + high)/2;
    until (high − low < ε mid);
    C := mid;
end.
```

Fig. 8.9 Algorithm for determination of capacity using a binary search

is feasible.

Since the computation of the functions $f_i(\cdot)$ in (8.9) is computationally expensive, we look for methods that do not require the computation of the partial derivatives of these functions.

8.4.1 Binary Search

A straightforward method of finding the capacity C comes from the second formulation given above, together with a trial-and-error application of Algorithm 8.3.1, say using a binary search. Algorithm 8.4.1, shown in Figure 8.9, computes the capacity using this strategy. This method has the major drawback that, when the value of *mid* is close to the capacity C, a large number of iterations is required for the detection of the convergence or divergence of Algorithm 8.3.1. Some heuristic

insight can be obtained regarding this behavior by looking at a one-dimensional situation where in (8.8) one of the rescheduling rates G_i is taken as independent variable and all the other ones expressed as funation of it, and introduced in the corresponding equation of (8.9), obtaining

$$\alpha_i S = G_i f_i(G_i),$$

or

$$\frac{G_i}{\alpha_i S} = \frac{1}{f_i(G_i)}. \tag{8.10}$$

The solution of (8.10) corresponds to the intersection of the curves $G_i/\alpha_i S$ and $1/f_i(G_i)$, and can be arrived at by the iterative process shown in Figure 8.10. The specific situation shown in this figure is obtained from the four node chain of Figure 5.1 operating under CSMA with idealistic perfect capture and uniform traffic pattern, in which G_3 was taken as the independent variable. The capacity is given by the value of S for which the line $G/\alpha_i S$ is tangent to the curve $1/f_i(G_i)$. It is apparent from Figure 8.10 that, for S larger than the capacity, the ratios $G^{(k+1)}/G^{(k)}$ start by decreasing, and then increase after some point. This behavior is verified in practice, and is the motivation for the divergence criterion used in Algorithm 8.3.1. From the same figure it is also apparent that, as S gets close to capacity, either from above or from below, the number of iterations required to recognize the convergence or divergence of the fixed-point iteration increases, and the rate of convergence decreases as we approach the solution. In the situation depicted, point P_2 is also a solution, and an iteration with starting point $G^{(0)} > G_2$ is seen to diverge. This behavior was also verified to occur in practice in the multidimensional case described by Equation (8.6).

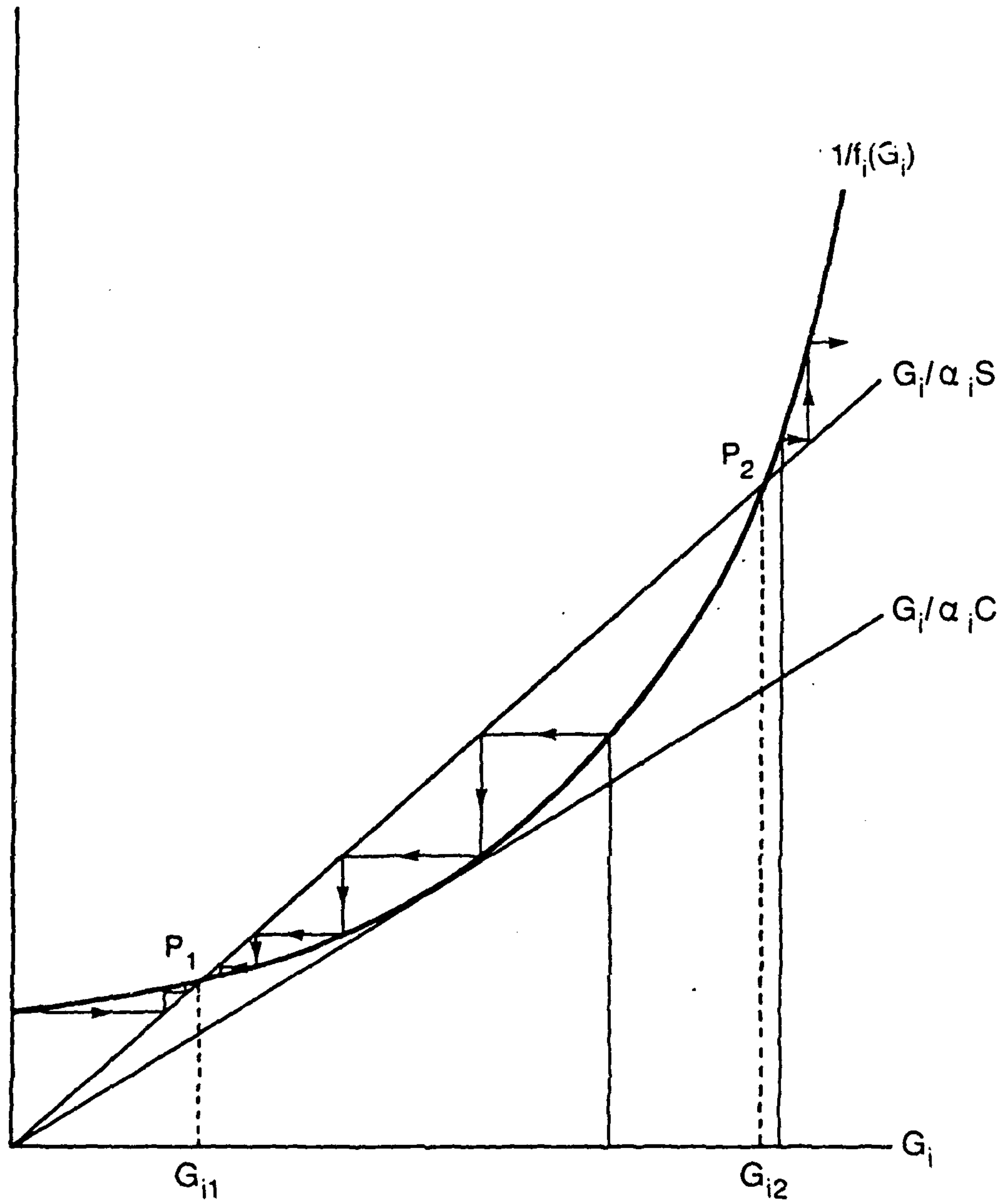


Fig. 8.10 Direct method of solution for desired throughpts

8.4.2 Parametric Method

An alternate approach comes from the first formulation given. Let us look at the set of constraints (8.8) as defining a line in the $(G_1, \cdots, G_L)$-space. To each point of this line there corresponds a value of S, given by any of the equations (8.9). The capacity C can be determined by constructing a family of hyperplanes indexed by some parameter g, finding the value $S(g)$ corresponding to the intersection of the line and the hyperplane, and maximizing by search over g. Formally, let $h(G_1, \cdots, G_L)$ be a linear functional defined by

$$h(G_1, \cdots, G_L) = \sum_{i=1}^{L} \beta_i G_i,$$

with $\beta_i \geq 0, i = 1, \cdots, L$. Let us consider points in the locus of $h(G_1, \cdots, G_L) = g$, where g is some (fixed) positive real number. By rewriting Equation (8.4) as

$$G_i = \frac{\alpha_i S}{f_i(\mathbf{G})}, \quad i \in \mathcal{L} \tag{8.11}$$

we obtain

$$h(G_1, \cdots, G_L) = S\, h\left(\frac{\alpha_1}{f_i(\mathbf{G})}, \cdots, \frac{\alpha_L}{f_L(\mathbf{G})}\right) \tag{8.12}$$

or, given that $h(\mathbf{G}) = g$,

$$S = \frac{g}{h\left(\frac{\alpha_1}{f_i(\mathbf{G})}, \cdots, \frac{\alpha_L}{f_L(\mathbf{G})}\right)}. \tag{8.13}$$

Introducing (8.13) in (8.11) we obtain

$$G_i = g\, \frac{\frac{\alpha_i}{f_i(\mathbf{G})}}{h\left(\frac{\alpha_1}{f_i(\mathbf{G})}, \cdots, \frac{\alpha_L}{f_L(\mathbf{G})}\right)}, \quad i \in \mathcal{L}, \tag{8.14}$$

Algorithm 8.3.3

begin
$k := 0;$
$\mathbf{G}^{(0)} := \mathbf{S}^{(0)};$
$S_i^{(0)} := G_i^{(0)} f_i(\mathbf{G}^{(0)})/\alpha_i, \qquad i \in \mathcal{L};$
$\beta^{(0)} := g/h(G_1^{(0)}/S_1^{(0)}, \ldots, G_L^{(0)}/S_L^{(0)});$
repeat
$k := k+1;$
$G_i^{(k)} := G_i^{(k-1)}\beta^{(k-1)}/S_i^{(k-1)}, \qquad i \in \mathcal{L};$
$S_i^{(k)} := G_i^{(k)} f_i(\mathbf{G}^{(k)})/\alpha_i, \qquad i \in \mathcal{L};$
$\beta^{(k)} := g/h(G_1^{(k)}/S_1^{(k)}, \ldots, G_L^{(k)}/S_L^{(k)});$
until $(\max_{i\in\mathcal{L}}\{S_i^{(k)}\} - \min_{i\in\mathcal{L}}\{S_i^{(k)}\} < \epsilon \min_{i\in\mathcal{L}}\{S_i^{(k)}\});$
$S := \beta^{(k)};$
end.

Fig. 8.11 Algorithm for parametric method of solution of throughput equations

which does not contain S, and can be solved using the fixed-point iteration algorithm shown in Figure 8.11. Note that the rescheduling rates $\mathbf{G}^{(k)}$ produced in the k-th iteration are such that $h(\mathbf{G}^{(k)}) = k$. After (8.14) is solved, the throughput S can be found from (8.13). The advantage of this formulation resides in that the functional $h(\cdot)$ can be chosen so that (8.14) is feasible for every $g \geq 0$. Whereas we do not have a formal proof of it, it has been the experience with all of the cases encountered that there exists at least one link i such that the locus defined by Equation (8.3) possesses points with the coordinate G_i arbitrarily large. Thus, by choosing $h(G_1, \cdots, G_L) = G_i$, the system (8.9) will be feasible for any $g \geq 0$. It has also been our experience that fixed-point iteration applied to (8.14) will converge in these circumstances. In the absence of information on which link i satisfies the above condition, one can choose $h(G_1, \cdots, G_L) = \sum_{r=1}^{L} G_i$, or any other functional in which the coefficients β_i [illegible] nonzero. Some choices of $h(\cdot)$

can speed up significantly the iteration process, but obtaining a good choice is very much a matter of intuition and of trial-and-error. In the case where Algorithm 8.3.3 is successively used for values of g that are close, the number of iterations required for the algorithm to converge can be substantially reduced by taking the initial rescheduling rates $\mathbf{G}^{(0)}$ of the iterative process for a new value of g to be the final value of the rescheduling rates at the completion of the algorithm for the previous value of g.

Fixed-point iteration applied to (8.14) has been seen to converge much faster than when applied to (8.6). We can again gain some insight into this behavior by considering a one-dimensional situation, corresponding again to the four node chain of Figure 5.1 under CSMA with idealistic perfect capture and uniform traffic pattern. Figure 8.12 shows the graph of Equation (8.14) when written for link 1, when we choose $h(\mathbf{G}) = G_3$, (i.e., when G_3 is taken to be the independent parameter g) and all other rescheduling rates are expressed as function of G_3. It is seen from the figure that the iteration converges for any value of g, although the number of iterations required for a given precision in the solution increases with g. However, for the values of g corresponding to the maximum of S, the number of iterations required is still much smaller than for the fixed point iteration used in the trial-and-error method of Algorithm 8.4.1. The situation shown in this figure is typical of the systems encountered.

The capacity C can be found by maximizing $S(g)$ over g. The function $S(g)$ is found in practice to be either monotonic increasing or unimodal, which facilitates the search for the maximum. For this purpose any standard method of maximizing functions of one variable can be used. In our implementation of the capacity computation, a quadratic curve fitting algorithm was employed ([Luen73]), with good results.

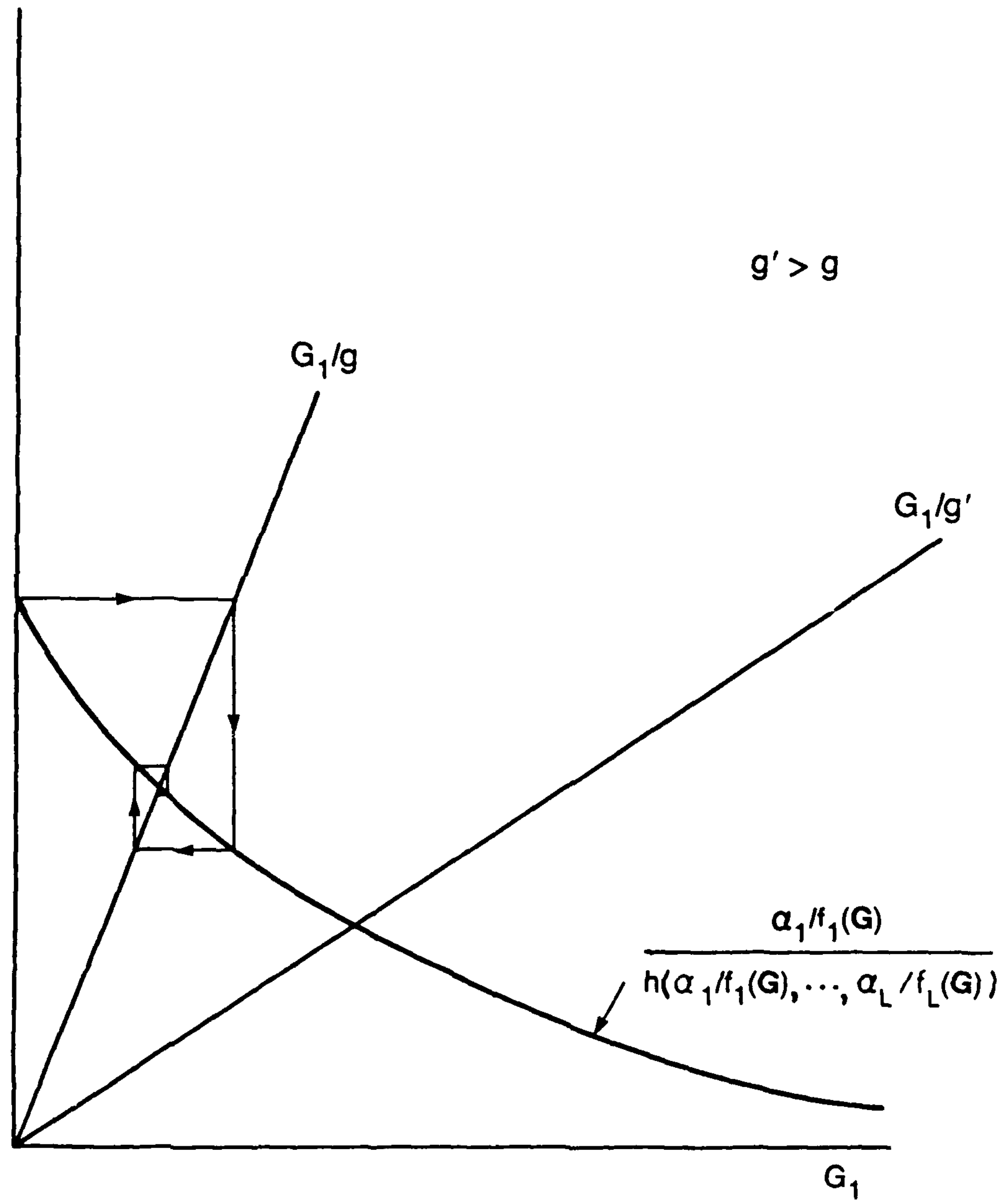


Fig. 8.12 Parametric method of solution for desired throughputs

8.5 Summary and Conclusions

We presented in this Chapter algorithms and data structures for the numerical evaluation of network capacity using the formalism developed in the previous Chapters. In Section 8.2 we considered the computation of the link throughputs given the channel access protocol, the *capture mode*, and the operating operating parameters (rescheduling rates and average message lengths) of the network. Most of this Section considered the more general protocols of Chapter 6, which do not lead to decoupling between the activity of the transmitters and the activity of the receivers. These protocols require the numerical setting up and solution of the balance equations of the processes involved. The enumeration of the state space of these processes was reduced to the traversal of a directed graph, and a breadth-first search (bfs) was considered for this purpose. We then studied some properties of the bfs enumeration of the state space for symmetric protocols, in particular those properties concerning the order in which states are enumerated. From *these properties* we derived efficient algorithms for the enumeration of state space. Such ordering properties do not exist for nonsymmetric protocols, and it was argued that the enumeration of the state space for a nonsymmetric protocol is more easily accomplished through the enumeration of the state space of the symmetrized version of the protocol (albeit at the expense of introducing additional transient states). Finally, we gave algorithms and data structures appropriate for a computer-implemented enumeration of the state space, setting up of the balance equations, solution of the systems of linear systems involved, and computation of throughput. We also gave similar results for the numerical computations involving protocols of class $\mathcal{D}$. In Section 8.3 we considered the problem of, given a set of link throughput requirements, solving for the network operating parameters that attain those requirements, for whose solution we presented a fixed-point iteration algorithm. In Section 8.4 we studied the problem of finding the capacity corresponding to an a *priori* given

traffic pattern. Two strategies were presented: (i) a trial-and-error binary search method, at each step of which the feasibility of a tentative value of the capacity is tested, and (ii) a parametric method that, given an arbitrary linear functional h of the rescheduling rates, finds the network throughput as a function of the value assigned to h, and then obtains the capacity by maximization over that value. In practice, the latter method was found to possess a superior performance.

Chapter 9

NUMERICAL APPLICATIONS

In this Chapter we apply the methodology developed in the previous Chapters to the study of the capacity performance of different channel access protocols, and of the influence on this performance of system parameters such as the type of signaling, the bit duration, and, in spread spectrum systems, the codeword length. For this purpose we shall consider a spread spectrum system with parameters typical of existing packet radio systems, and a narrow band system with the same parameters as the spread spectrum system except for the coding. We consider a number of parametric topologies (rings, chains, and stars) and randomly generated topologies. A number of Busy Tone protocols, the Carrier Sense Multiple Access protocol, and two ALOHA protocols are considered, in both narrowband and spread spectrum environments. Section 9.1 specifies the spread spectrum and the narrowband systems considered. It also describes the noise and the capture model, the topologies, traffic patterns, and access protocols considered. Section 9.2 presents the numerical results. Even though only small networks can be accommodated due to the fast growing computational complexity of the analysis, the results obtained allow some insight to be gained. In particular, they establish a ranking of the protocols

in terms of their capacity performance, and illustrate the tradeoff achieved by the different protocols and signaling methods in controlling collisions at the receivers versus allowing the transmitters wide access to the channel.

9.1 System Description

9.1.1 Spread Spectrum System

We consider a system with the parameters given in [Fral75], [Kahn78], [Behr82]. The chip duration in this system is $T_c = 78.125$ nsec. Two different codeword lengths are used, $N_c = 32$ chips/bit and $N_c = 128$ chips/bit. To these codeword lengths there correspond two different bit durations, $T_b = 2.5\,\mu$sec and $T_b = 10.0\,\mu$sec, with data rates of 400 kbps and 100 kbps, respectively.

All transmitters are assumed to have the same transmitted power. The transmitted power, thermal noise density and unit of distance are taken to be such that, for the shorter bit duration $T_b = 2.5\,\mu$sec, the resulting signal-to-noise ratio at unit distance in the absence of multiuser interference $(S/N)_{0,u} = A^2 T_b/\eta_0$ is 11 db (see Equation (I.3)). To this signal-to-noise ratio there corresponds, from Equation (I.4), a probability of bit error in the absence of multiuser interference $P_e = 1.94 \times 10^{-4}$ at unit distance. The signal-to-noise ratio for the longer bit duration $T_b = 10.0\,\mu$sec is four times the value for $T_b = 2.5\,\mu$sec, or roughly 17 dB, to which corresponds a probability of bit error $P_e = 6.42 \times 10^{-13}$ at unit distance in the absence of multiuser interference. It is assumed that, due to the choice of preamble codes with good correlation properties, an unlocked receiver locks with probability one onto the preambles of new packets destined to it. Packets are taken to contain an average number of 1000 bits.

9.1.2 Narrowband System

We consider the narrowband system obtained from the spread spectrum system of Section 9.1.1 by removing the coding (for example, by taking all chips to have amplitude +1). Thus again we consider two different bit lengths, $T_b = 2.5\,\mu$sec and $T_b = 10.0\,\mu$sec, to which there correspond signal-to-noise ratios at unit distance $(S/N)_{0,u} = 11$ dB and $(S/N)_{0,u} = 17$ dB, with probabilities of bit error $P_e = 1.94 \times 10^{-4}$ and $P_e = 6.42 \times 10^{-13}$, respectively.

In the cases examined we shall consider a situation of zero capture, in which any packet overlay at a receiver causes the destruction of both packets at that receiver. This behavior constitutes a good approximation whenever the levels of the interfering signals at a receiver are not much smaller than the level of the desired signal, as discussed in Section I.5.

9.1.3 Noise and Capture Model

We assume the noise model of Section I.4, and the capture model of Section 5.3.2.2 (which includes, as a particular case, the zero capture of narrowband systems). In order to apply Equation (I.4) to determine the probability of bit error one needs to know the power of the received signal as a function of the distance to the transmitter, that is, the propagation law. A satisfactory model consists of taking the received power to vary as $R^{-\alpha}$, where R is the distance to the transmitter and α is a constant between 3 and 4 ([Fral75]). In the following computations we take $\alpha = 3$.

For narrowband systems, the probability of bit error if no packet overlap occurs at the receiver is $P_e = Q\left(\sqrt{(S/N)_0}\right)$ and with $(S/N)_0 = R_0^{-3}(S/N)_{0,u}$, where R_0 is the distance from the desired transmitter to the receiver. If a packet overlap occurs,

the probability of bit error is taken to be 1/2. In the latter case, we take the probability of success of such packet to be zero. For spread spectrum systems, we take the probability of bit error due to interference other than "vulnerable window" collisions to be given by Equation (I.4). In this equation, and given the assumptions of equal transmitted power and propagation as R^{-3}, we can write $P_k/P_0 = (R_k/R_0)^{-3}$, where R_k is the distance from the k-th interfering transmitter to the transmitter. If we let $(S/N)_c \triangleq A^2T_c/\eta_0$ be the signal-to-noise ratio corresponding to a received pulse of duration equal to the chip duration, then $(S/N)_0 = N_c(S/N)_c$, and the probability of bit error given by Equation (I.4) becomes

$$P_e = Q\left(\sqrt{\frac{N_c(\frac{S}{N})_c}{1 + (\frac{S}{N})_c \sum_{k=1}^{p+q} (\frac{R_k}{R_0})^{-3}}}\right),$$

which shows the improvement in bit error rate obtained when the bit duration is changed by varying the length of the codewords.

9.1.4 Topologies

We consider the following topologies: chain topologies, ring topologies, star topologies, and a small number of nonparametric topologies. The nonparametric topologies considered are three randomly-generated topologies and a centered-square topology, shown in Figure 9.1. For the purposes of received power, all links in the star, chain, and ring topologies, and the links forming the sides of the square, in the centered square topology, are taken to have unit length. The random networks were generated by randomly placing six nodes in a square whose side has a length of two units. The coordinate of the nodes were obtained as realizations of independent random variables uniformly distributed within the length of the side

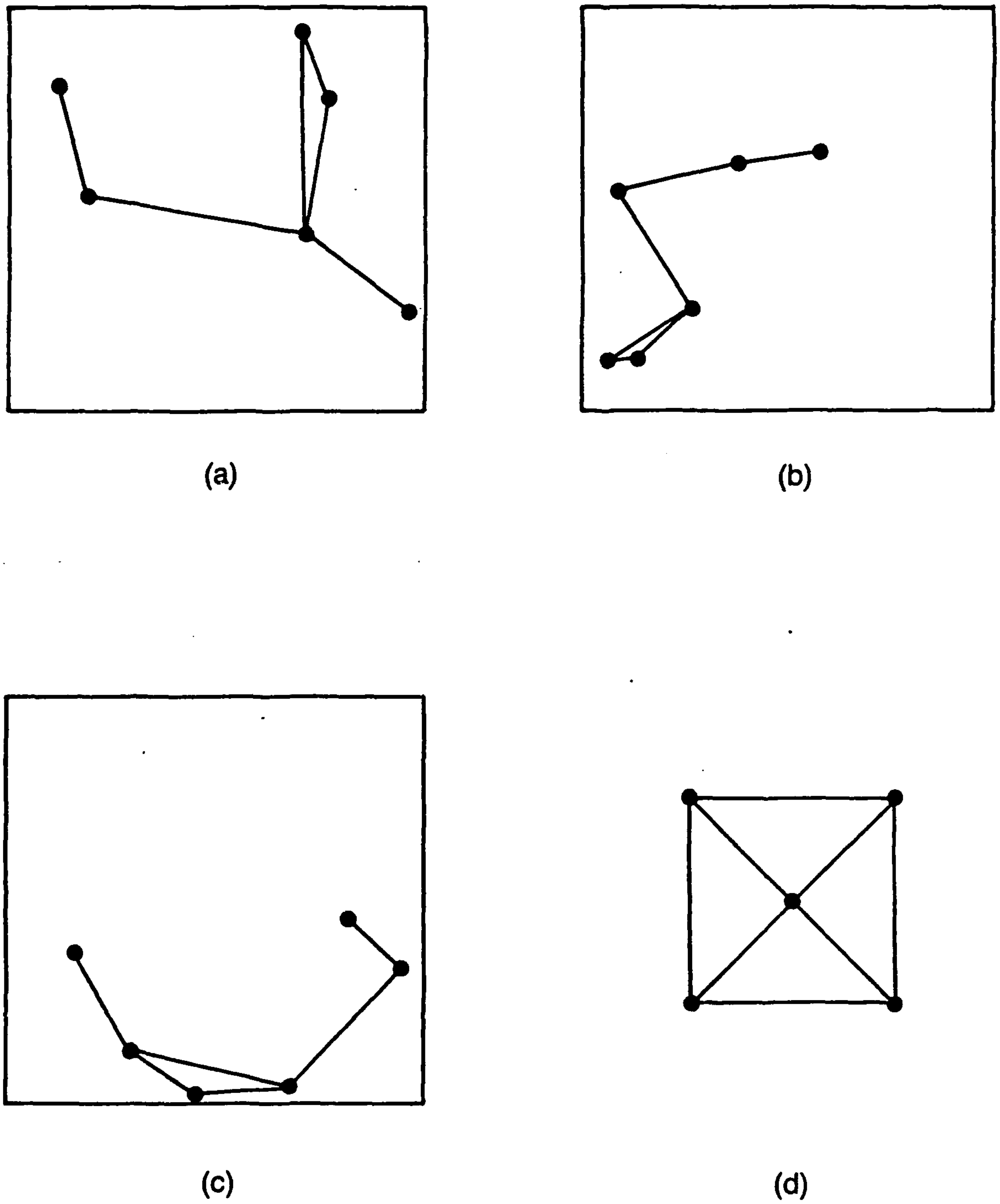


Fig. 9.1 Nonparametric topologies considered in examples

of the square. The connectivity was determined from the smallest hearing radius that makes the network connected.

The rapid growth of the storage and computation requirements of the analysis as a function of the number of network elements limits the direct application of the analysis to small networks. However, the networks that we consider represent well the different aspects of multihop networks: on one hand the multihop nature, represented by the chain and ring network, and on the other hand the varying degree of connectivity, given by the star networks of different sizes, and by the centered square network. The random networks combine both aspects.

9.1.5 Traffic Patterns

Two distinct types of traffic pattern were considered: (i) uniform nearest-neighbor traffic (NNB), in which all links carry the same traffic, and (ii) uniform end-to-end traffic (ETE) with minimum-hop routing. The link traffic requirements in the latter case are determined by computing the shortest paths between the source and the destination nodes, and then assigning flows to the links in those paths. Whenever at a node in the shortest path there are more than one shortest paths to the destination, the outgoing traffic is split evenly among those paths. In the case of uniform link traffic we set $\alpha_i = 1$ for all links $i \in \mathcal{L}$, where α_i is the traffic pattern coefficient figuring in Equation (8.8). In this case the capacity C, defined by Equation (8.7), represents the maximum link throughput. In the case of uniform end-to-end traffic the coefficients α_i are computed by assigning to every pair of nodes an end-to-end traffic of $1/N(N-1)$, where N is the number of nodes. The coefficient α_i for link i is set equal to the link traffic resulting from this end-to-end assignment. In this way, the capacity C represents the maximum value of the sum of the end-to-end traffics that is supported by the network under minimum-hop routing and uniform end-to-end requirements.

For the star and ring topologies both traffic patterns lead to uniform link traffic, although with different capacities. Let C_n and C_e be the capacities under the NNB and ETE traffic patterns, respectively. In terms of the NNB capacity, the total internal traffic equals LC_n, whereas in terms of the ETE capacity it equals $\overline{n}C_e$, where $\overline{n}$ is the average number of hops traveled by a message. Thus

$$C_e = \frac{L\,C_n}{\overline{n}}.$$

The following results are easily obtained:

(i) *Ring with $2n+1$ nodes:*

$$C_e = \frac{4(2n+1)}{n+1}\,C_n$$

(ii) *Ring with $2n+2$ nodes:*

$$C_e = \frac{4(2n+1)}{n+1}\,C_n$$

(iii) *Star with n nodes (including node at center of star):*

$$C_e = n\,C_n.$$

In the examples that follow, we shall consider the ETE capacity for the star network, and the NNB capacity for the ring network. For some of the situations involving the remaining topologies both traffic patterns will be considered.

9.1.6 Protocols

For the narrowband systems and the spread spectrum with bit-homogeneous codes, the following protocols will be considered: C-BTMA, ID-BTMA, LD-BTMA,

CSMA, D-ALOHA, and ALOHA. ID-BTMA is not directly feasible in some of these system configuration and we include it here for comparison purposes only. For spread spectrum systems with bit-changing codes, in which activity sensing is not feasible, we shall consider only D-ALOHA and ALOHA.

9.2 Results

The results obtained are presented in Figures 9.2 through 9.42. A label with four fields was appended to each figure in order to simplify the identification of the situation the figure refers to. The meaning of these fields are as follows:

(i) Topology: ring, star, chain, or nonparametric topology;

(ii) Traffic pattern: uniform end-to-end (ETE), or uniform nearest-neighbor (NNB);

(iii) Signaling/capture: narrowband with zero capture (NBZC), narrowband with idealistic perfect capture (NBPC), spread spectrum with bit-changing codes (SSBBCC), or spread spectrum with bit-homogeneous codes (SSNBBCC);

(iv) Bit duration/code assignment: all the bit-homogeneous systems considered use the longer bit duration; in the case of these systems, this field indicates whether a transmitter-directed (TXDIR), receiver-directed (RCDIR), or uniform (UNIF) code assignment is used. In the case of narrowband systems, or spread spectrum systems with bit-changing codes, this field indicates whether the longer bit duration (LG) or short bit duration (SH) is used.

The results for ring networks are contained in Figures 9.2 through 9.12. The capacity of the narrowband systems are shown in Figures 9.2 (for the longer bit duration $T_b = 10\mu$sec) and 9.3 (for the shorter bit duration $T_b = 2.5\mu$sec). The system with longer bit duration possesses higher capacity, due to the lower bit error

rate, which can be considered negligible for all purposes. The relative performance of the two systems can be better seen in Figure 9.10. In order to see the influence of the collisions that take place after a packet transmission starts we show in Figure 9.4 the hypothetical situation of idealistic perfect capture. Here a node cannot lock onto a new packet as long as some of its neighbors are active, but is guaranteed successful reception after it is locked onto, irrespective of the activity of any other nodes (as long as the destination node does not switch to transmission during the packet's reception, in the protocols that so allow). As expected, the capacity of the Busy Tone protocols remains unchanged with respect to the situation of zero capture and low bit error rate, since the packet losses that are averted by the idealistic capture are already prevented by the Busy Tone protocols in the case of zero capture. The improvement in performance due to the perfect capture for ALOHA is small, the cause being that a large fraction of the packet loss is due to the destination node of a packet switching to transmit mode during the reception of the packet. CSMA and D-ALOHA show a significant improvement over the situation of zero capture. Figures 9.2 through 9.4 reveal that for the narrowband system the ordering of the protocols in terms of their performance is (i) C-BTMA, (ii) ID-BTMA, (iii) LD-BTMA, (iv) CSMA, (v) ALOHA, (vii) ALOHA, with CSMA and D-ALOHA having similar performances. We shall see this ordering to be maintained for narrowband systems under other topologies. The fact that C-BTMA performs consistently as well or better than ID-BTMA and LD-BTMA shows that in the narrowband systems the collision-free operation of C-BTMA outweighs the smaller blocking of ID-BTMA and LD-BTMA, which allow collisions, as discussed in Chapter 3.

Figures 9.5 and 9.6 show the performance of D-ALOHA and ALOHA in a spread spectrum system with bit-changing codes, for both the longer (Figure 9.5) and shorter (Figure 9.6) bit durations. The decrease in performance of the system with the shorter bit duration over the one with the longer duration is due now

to two effects: (i) higher bit error rate, due to the lower signal-to-thermal-noise ratio, and (ii) decreased protection against multiuser interference, due to the smaller codeword length. The spread spectrum system with longer bit durations presents an improvement over the narrowband idealistic perfect capture system that results from the capability that an unlocked node has in the former system of locking onto a new packet destined to it in the presence of neighboring activity, rather than having to wait for an idle channel before being able to lock onto a new packet, as in the latter. Figures 9.7 through 9.9 present results for the spread spectrum systems with bit-homogeneous codes in the situation of lower bit error rates ($P_e = 6.42 \times 10^{-13}$) and the uniform, receiver-directed, and transmitter-directed code assignments. The collisions resulting from the timing of interfering signals with the same code waveforms falling within the "vulnerable window" of the desired signal cause a degradation on the performance of these systems relative to the systems with bit-changing codes. In order to bring out the effect of these collisions, which is inversely proportional to the length of the codewords, we consider a system with the shorter bit duration ($N_c = 32$), but with a fourfold increase in the transmitted power in order to achieve the same probability of bit error. As expected, C-BTMA performs identically in the three cases, and identically to the way it performs in the narrowband system with the same probability of bit error, since it is collisions-free. Contrarily to the situation on the narrowband systems, ID-BTMA and LD-BTMA perform occasionally better than C-BTMA. This improvement in the performance of these protocols is due to (i) the fact that now an unlocked receiver can lock onto a new packet in the presence of neighboring activity, without having to wait for an idle channel, and thus a packet destined to an unlocked node is locked onto with probability one (and not collided with also with probability one), and (ii) the smaller number of nodes blocked by the busy tone under ID-BTMA and LD-BTMA allows higher communication concurrency, without the introduction of additional

packet loss. CSMA offers now a marked improvement over D-ALOHA, contrarily to the situation in the narrowband systems. The better performance of CSMA can be attributed to its avoiding the situation where a given unlocked node starts transmitting to a destination node which is active, thus wasting a transmission. As expected, the ordering of the bit homogeneous systems in descending order of performance is (i) transmitter-assigned codes, (ii) receiver-directed codes, and (iii) uniform codes, as shown in Figure 9.11 for CSMA, D-ALOHA, and ALOHA. In Figure 9.12 we show the comparative performance of the system with uniform codes and the narrowband system with the same probability of bit error.

Figures 9.13 through 9.17 show results for chain networks with uniform nearest-neighbor traffic. These results show that the performance of the different protocols has the same behavior with respect to the different system parameters as in the ring topologies. Figure 9.18 through 9.22 show results for the uniform end-to-end traffic pattern. Under this traffic pattern the links closer to the center of the chain carry more traffic than those farther away. From these figures one sees that the relative performance of the different channel access protocols is insensitive with respect to the type of traffic pattern.

Figures 9.23 through 9.29 show the performance of star networks with uniform end-to-end traffic. In these networks all busy tone protocols perform identically. The performance of these protocols is not shown for the spread spectrum bit-homogeneous systems, since it is identical to that of the corresponding narrowband system with the same probability of bit error. The ordering of the protocols on the basis of their performance is the same as for the ring networks.

Figures 9.30 through 9.41 show the performance of the different protocols in the nonparametric topologies. Each graph shows, from left to right, the performance of the topologies of Figure 9.1 in the order of their appearance in that Figure from top

to bottom. Figure 9.30 through 9.36 give results for the uniform nearest-neighbor traffic pattern. The relative performance of the access protocols is seen to be similar to the performance exhibited in the other topologies examined. Figures 9.37 through 9.41 show results for the case of uniform end-to-end traffic pattern. It is again seen that the relative performance of the channel access protocols is relatively unaffected by the type of traffic pattern. The influence of systems parameters such as signal-to-noise ration and type of signaling is also similar to that observed in the other topologies examined.

9.3 Summary and Conclusions

This Chapter presented capacity results for a number of topologies, channel access protocols, and capture modes. Section 9.1 described the spread spectrum and the narrowband systems considered. It also described the noise and the capture model, the topologies, traffic patterns, and access protocols considered. Section 9.2 presented the numerical results. These results establish a ranking of the protocols in terms of their capacity performance, and illustrate the tradeoff achieved by the different protocols and signaling methods in controlling collisions at the receivers versus allowing the transmitters wide access to the channel.

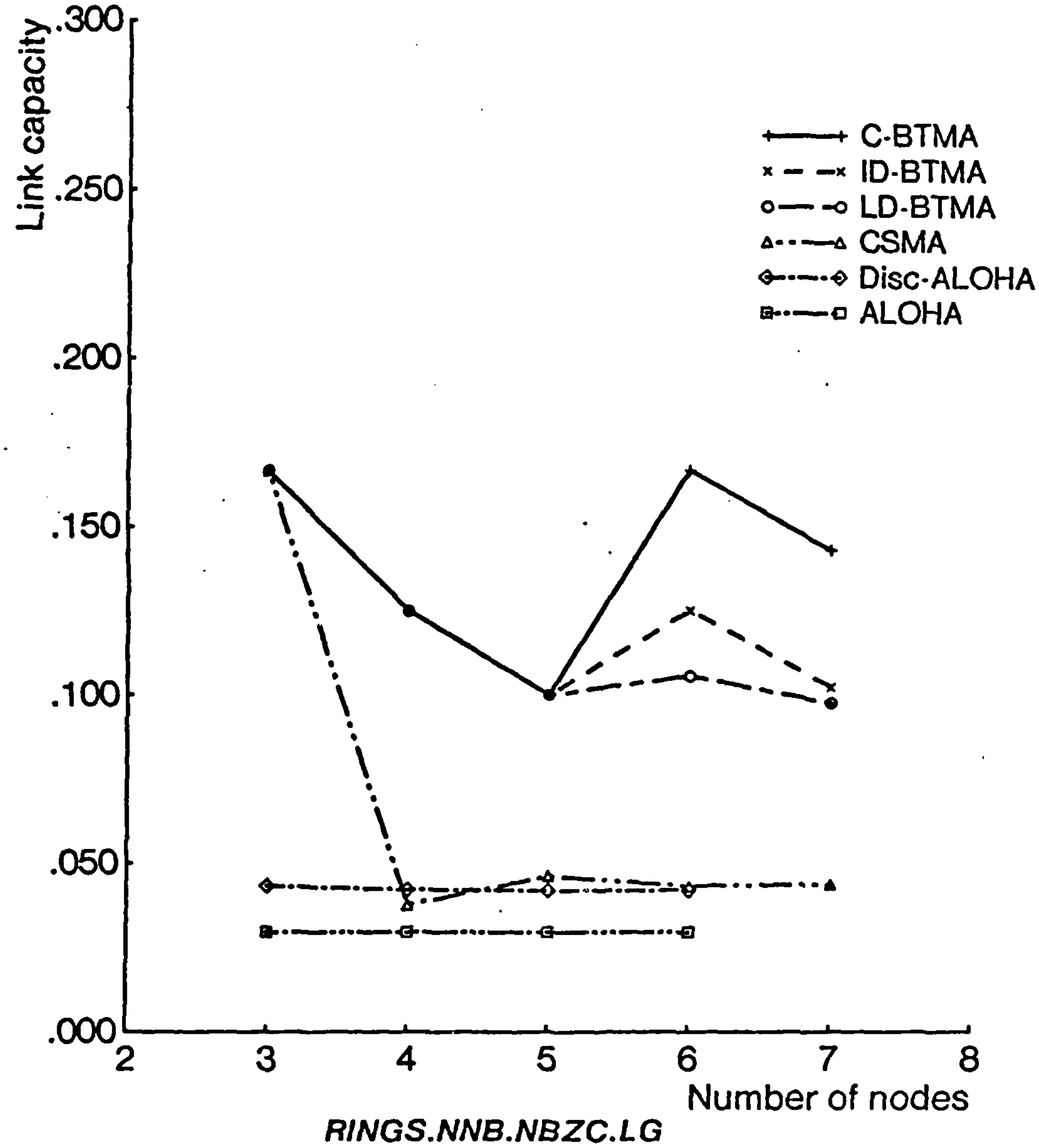


Fig. 9.2 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and long bit duration

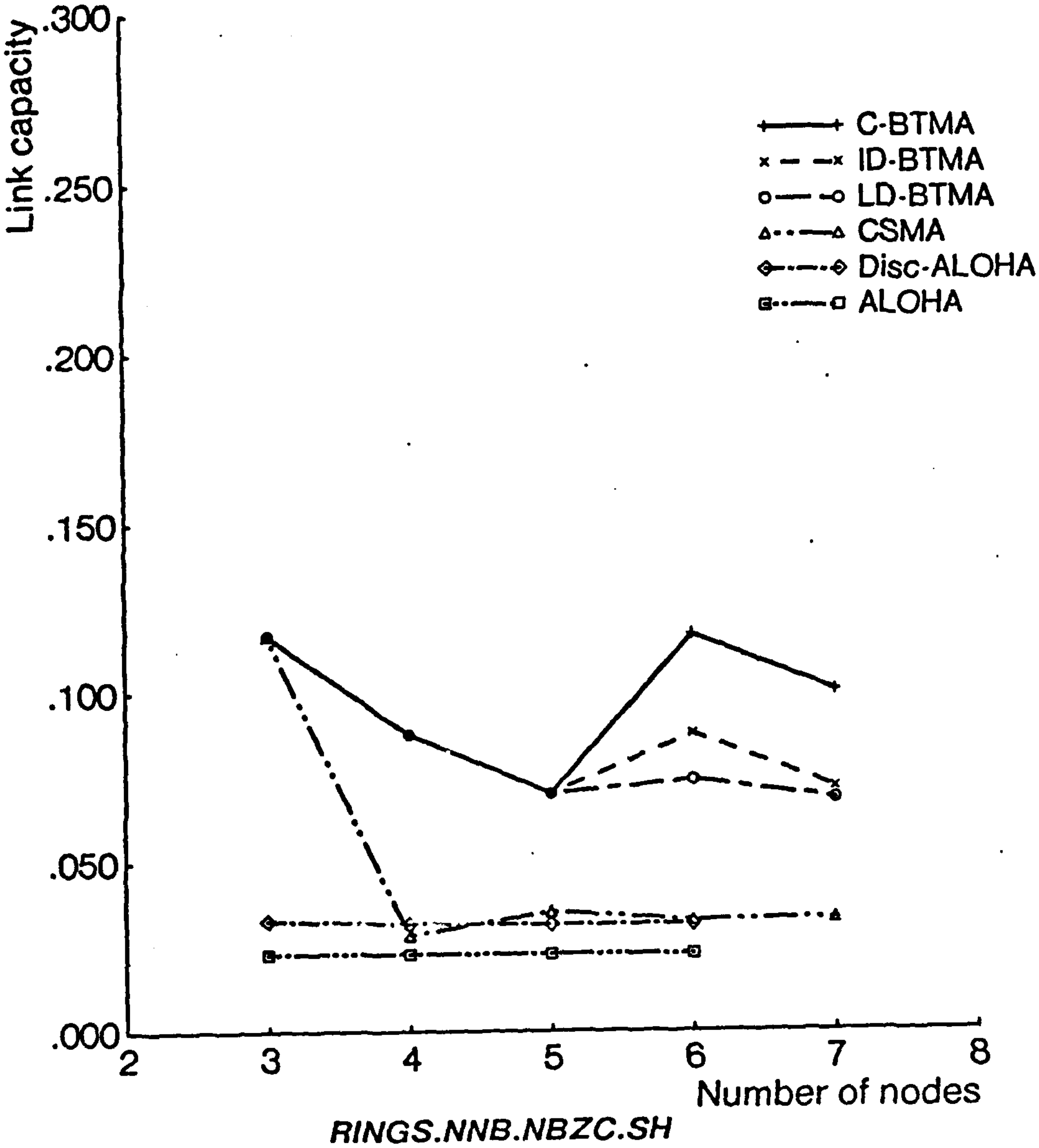


Fig. 9.3 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and short bit duration

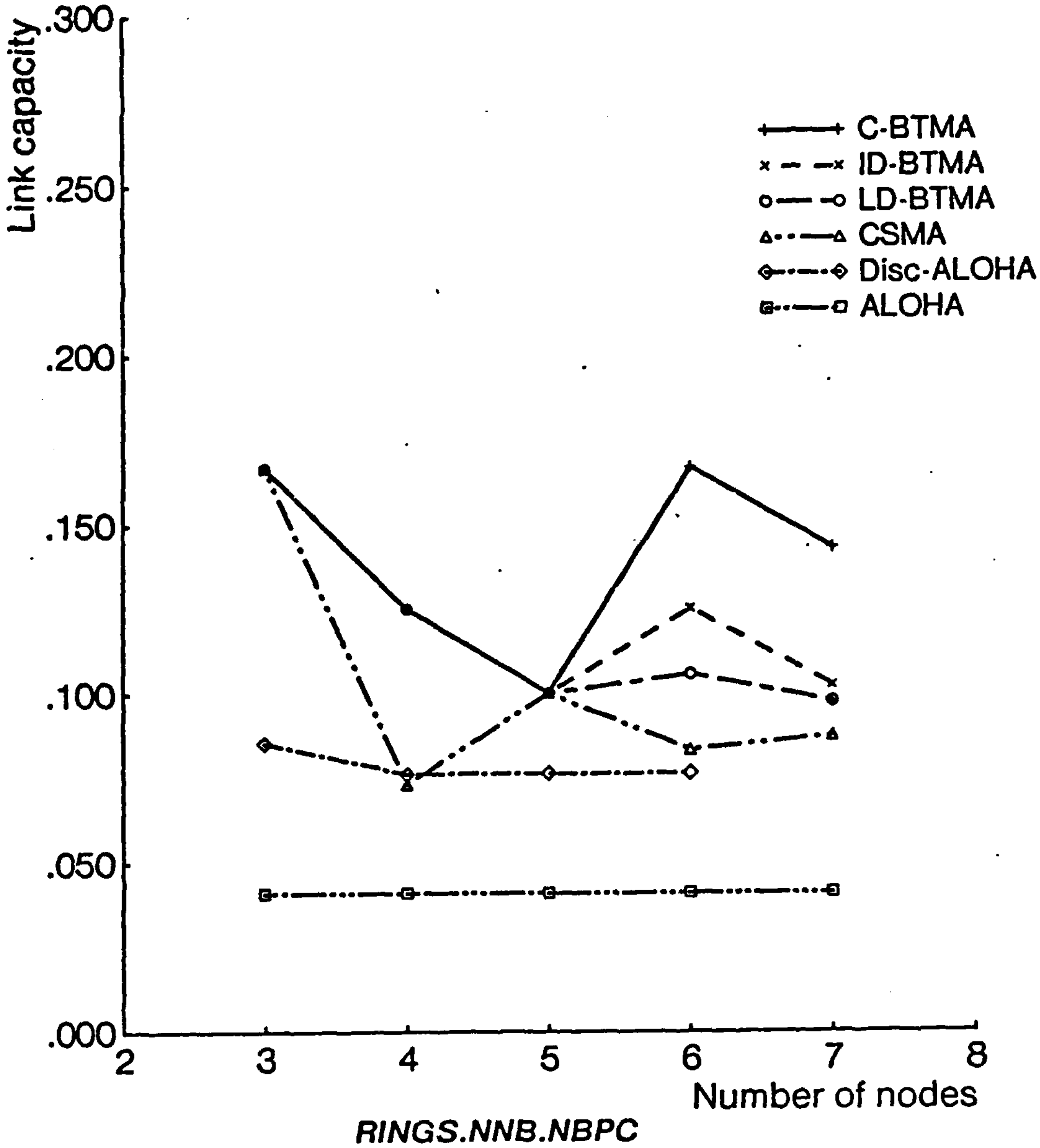


Fig. 9.4 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a narrowband system with idealistic perfect capture

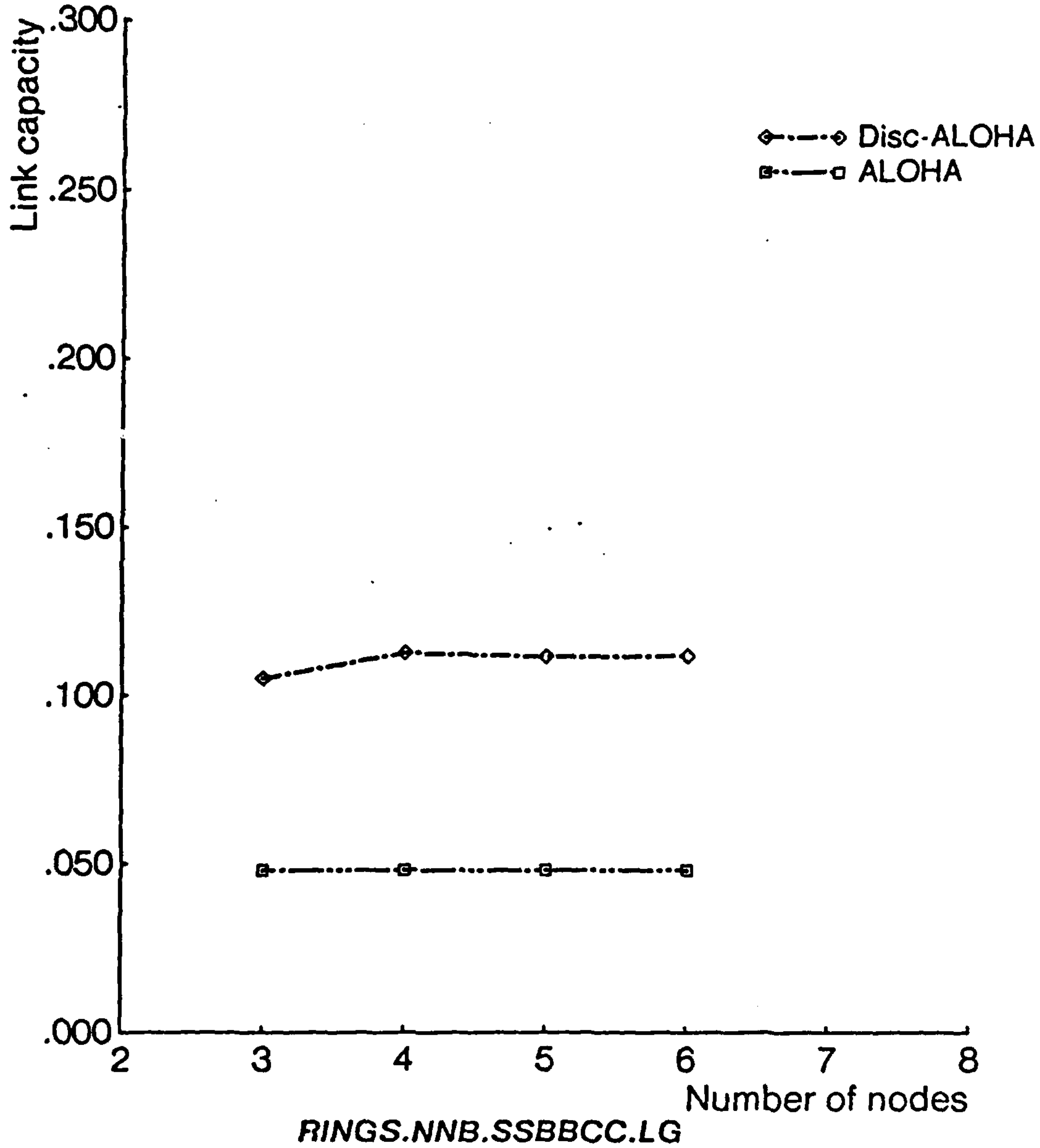


Fig. 9.5 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and long bit duration

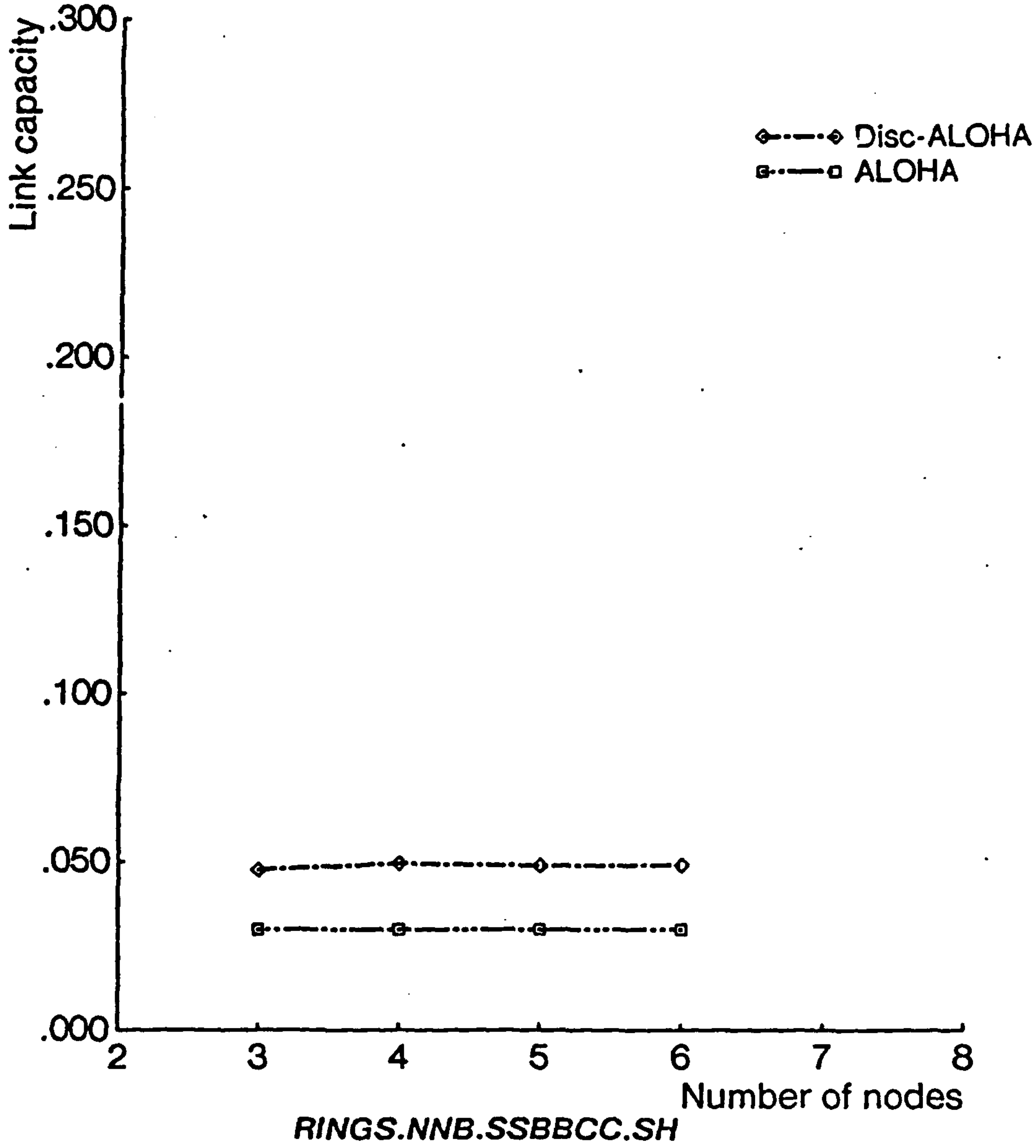


Fig. 9.6 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and short bit duration

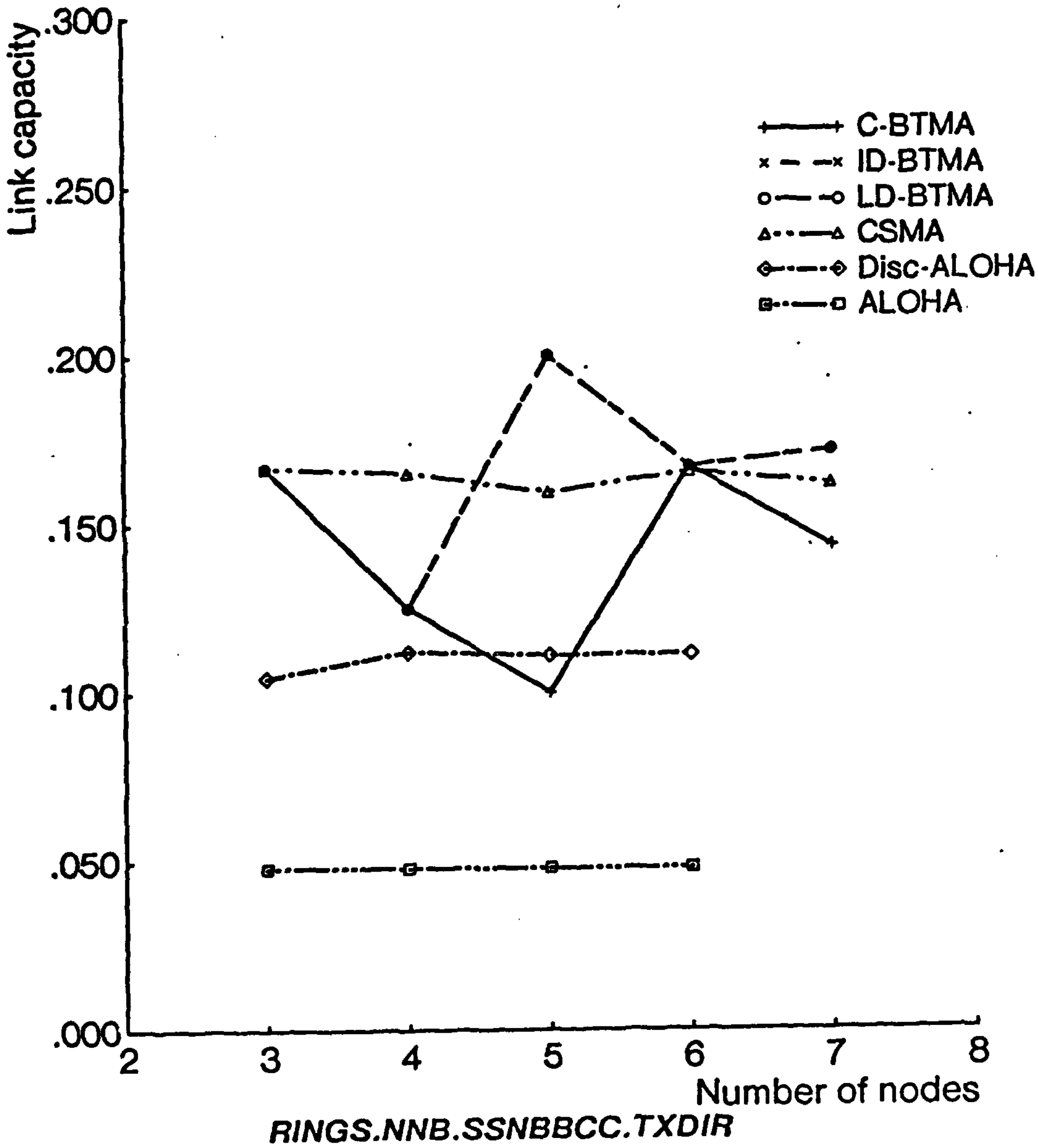


Fig. 9.7 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous transmitter-directed codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

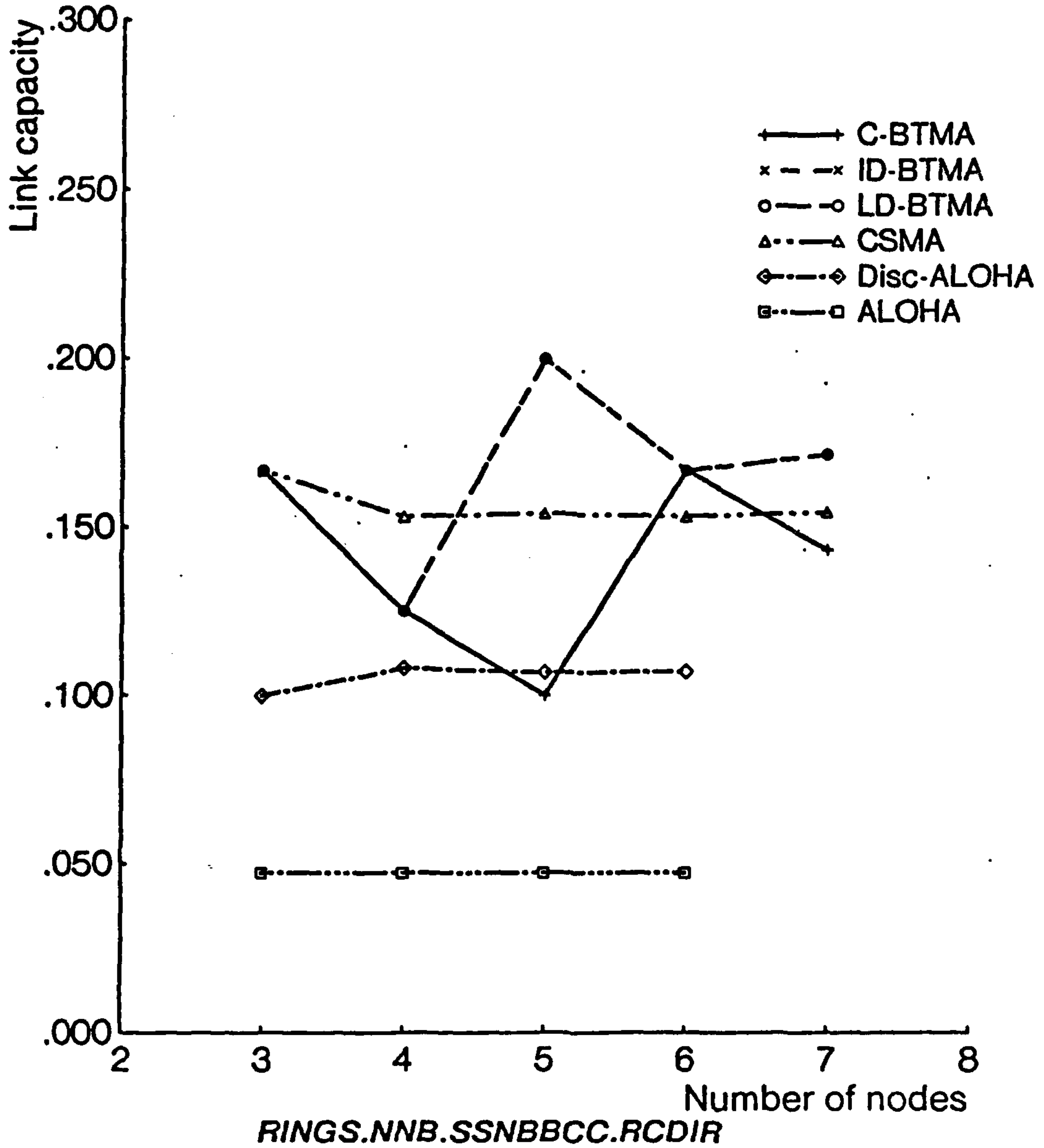


Fig. 9.8 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous receiver-directed codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

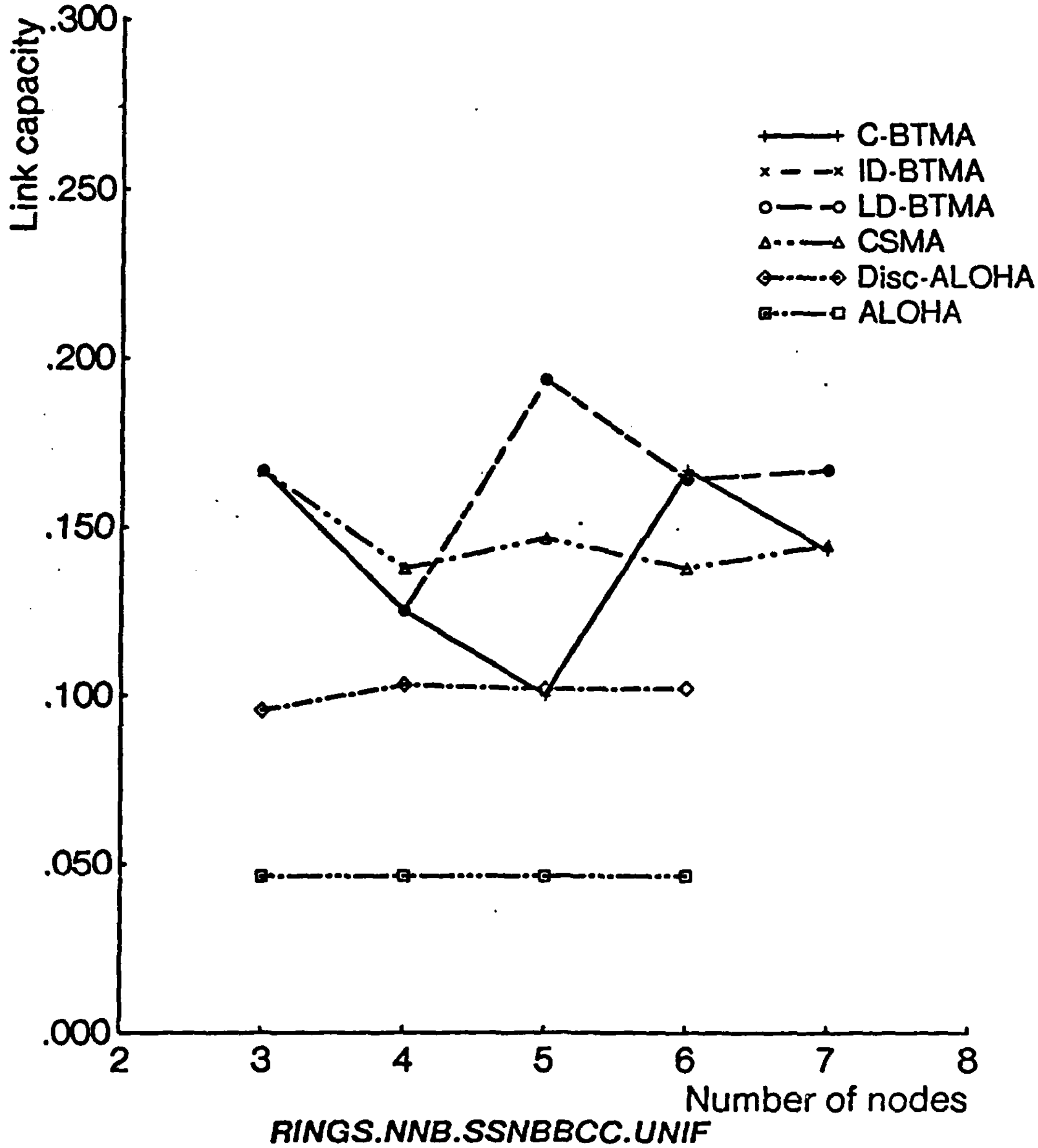


Fig. 9.9 Link capacity for ring topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous uniform codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

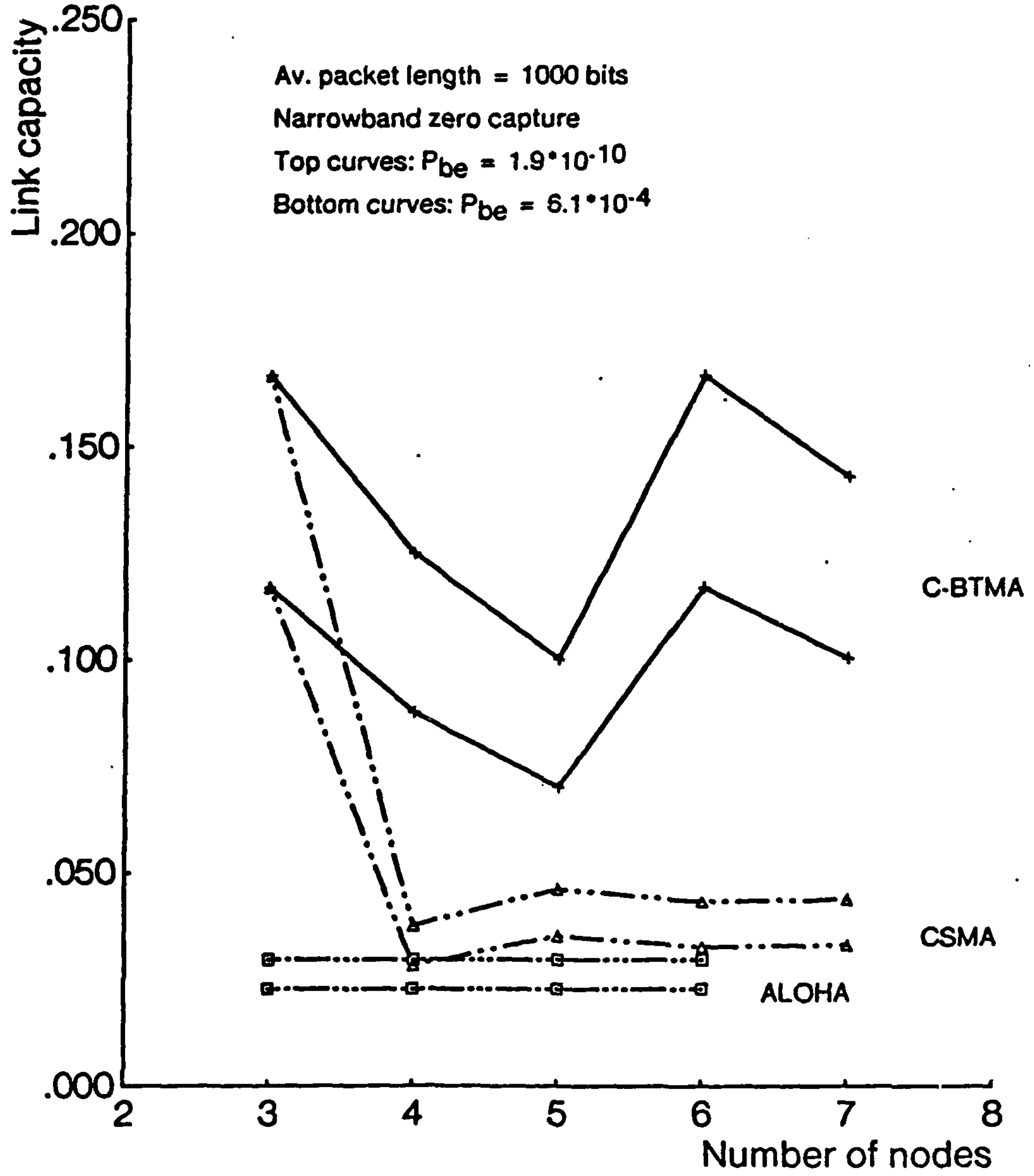


Fig. 9.10 Comparison of link capacity for long and short bit lengths, in a ring topology with uniform nearest-neighbor traffic and narrowband zero capture

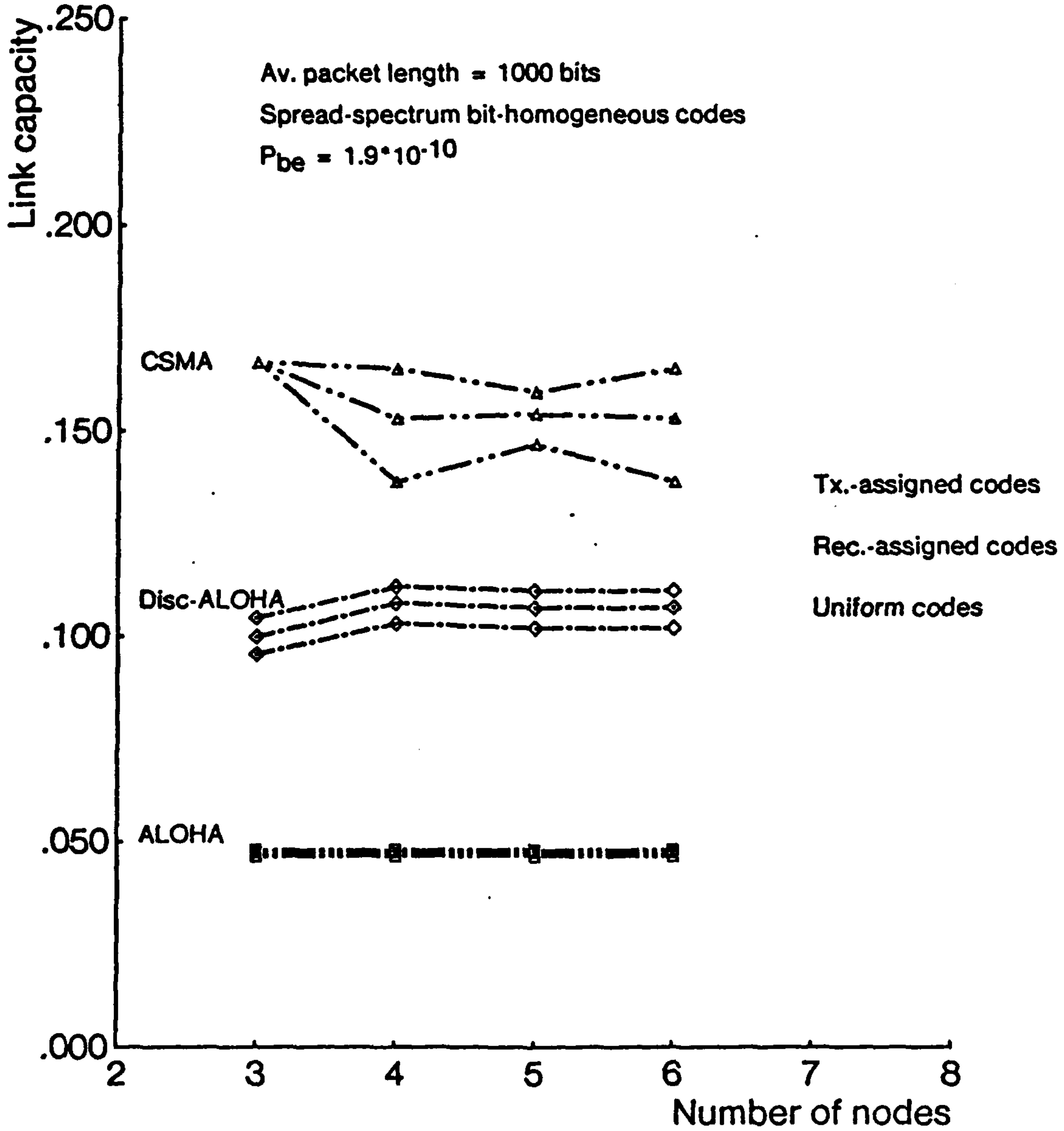


Fig. 9.11 Comparison of link capacity for the transmitter-assigned, receiver-assigned, and uniform code assignments, in a ring topology with uniform nearest-neighbor traffic and spread-spectrum signaling with bit-homogeneous codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

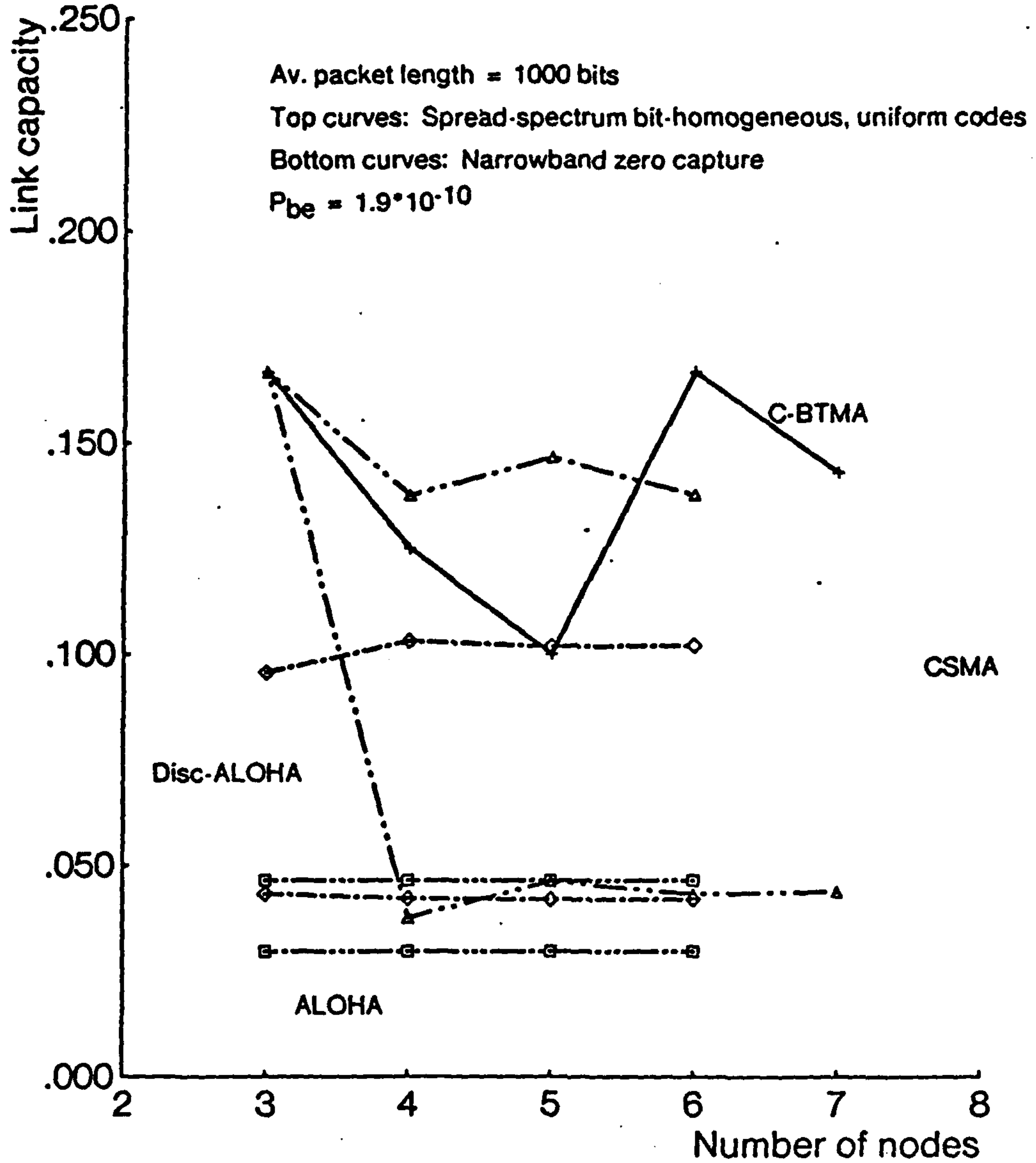


Fig. 9.12 Comparison of link capacity for a spread-spectrum bit-homogeneous system with uniform codes, and a narrowband zero capture system, for a ring topology with uniform nearest-neighbor traffic and long bit duration (see text for details on spread-spectrum system

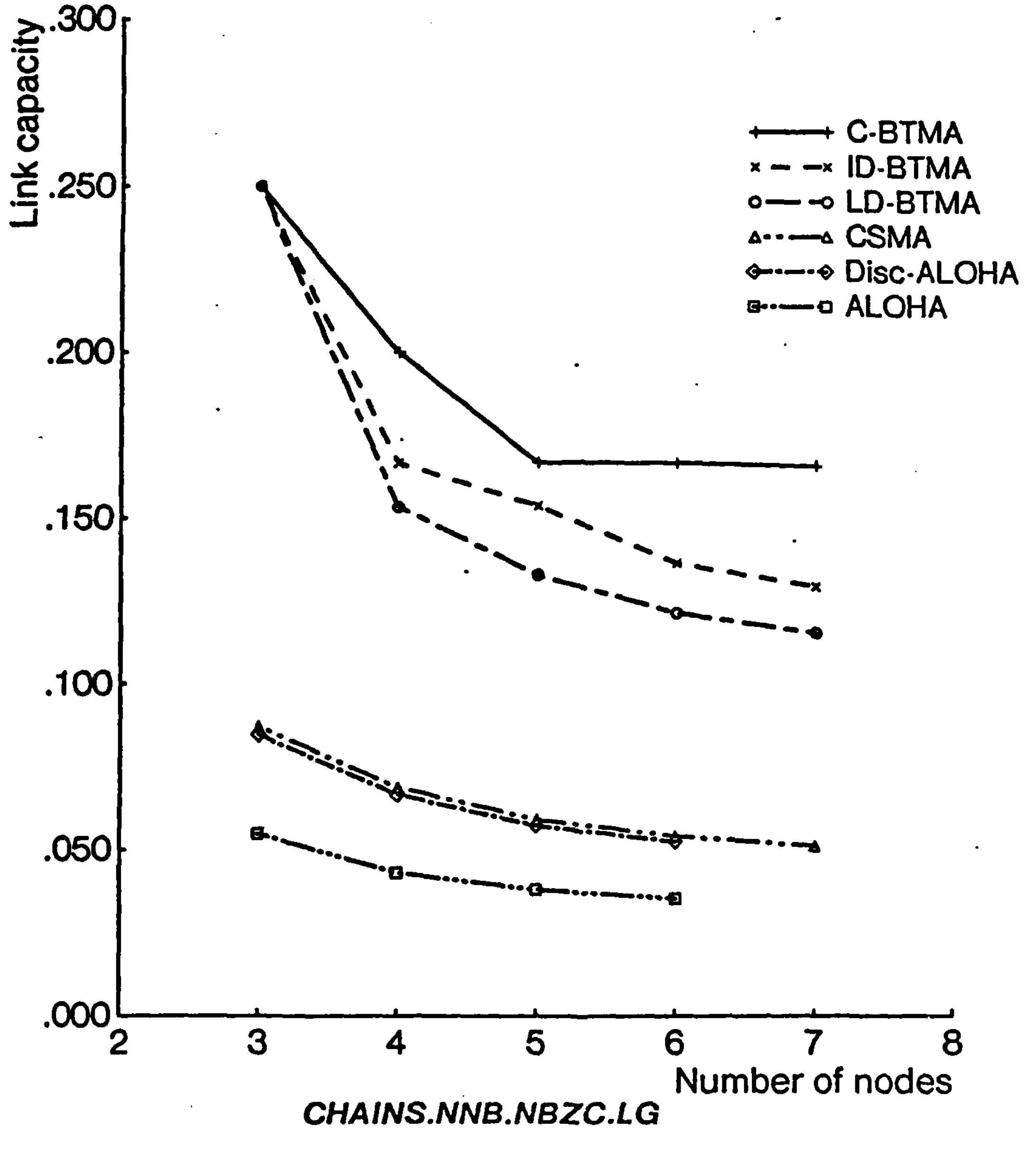


Fig. 9.13 Link capacity for chain topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and long bit duration

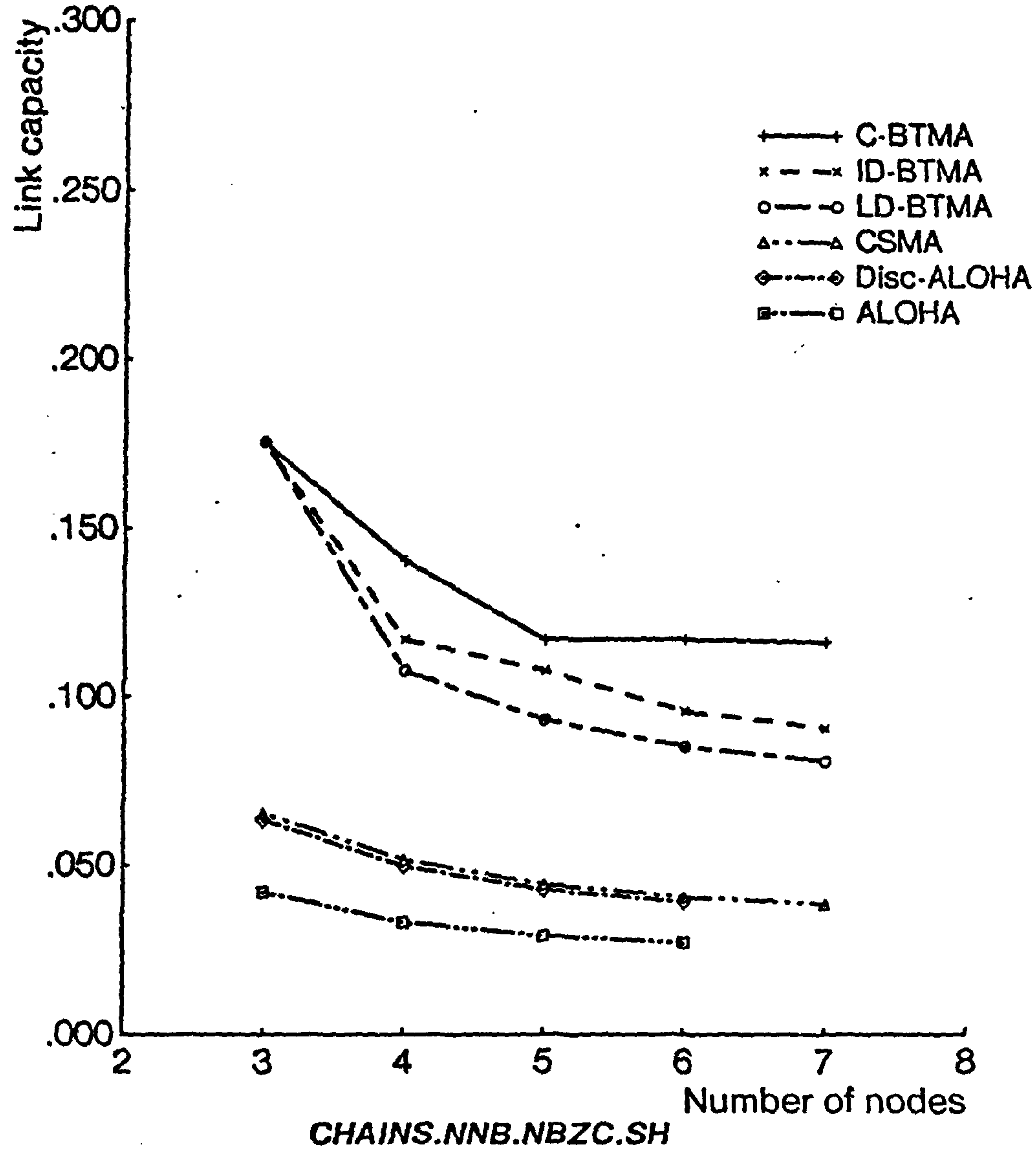


Fig. 9.14 Link capacity for chain topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and short bit duration

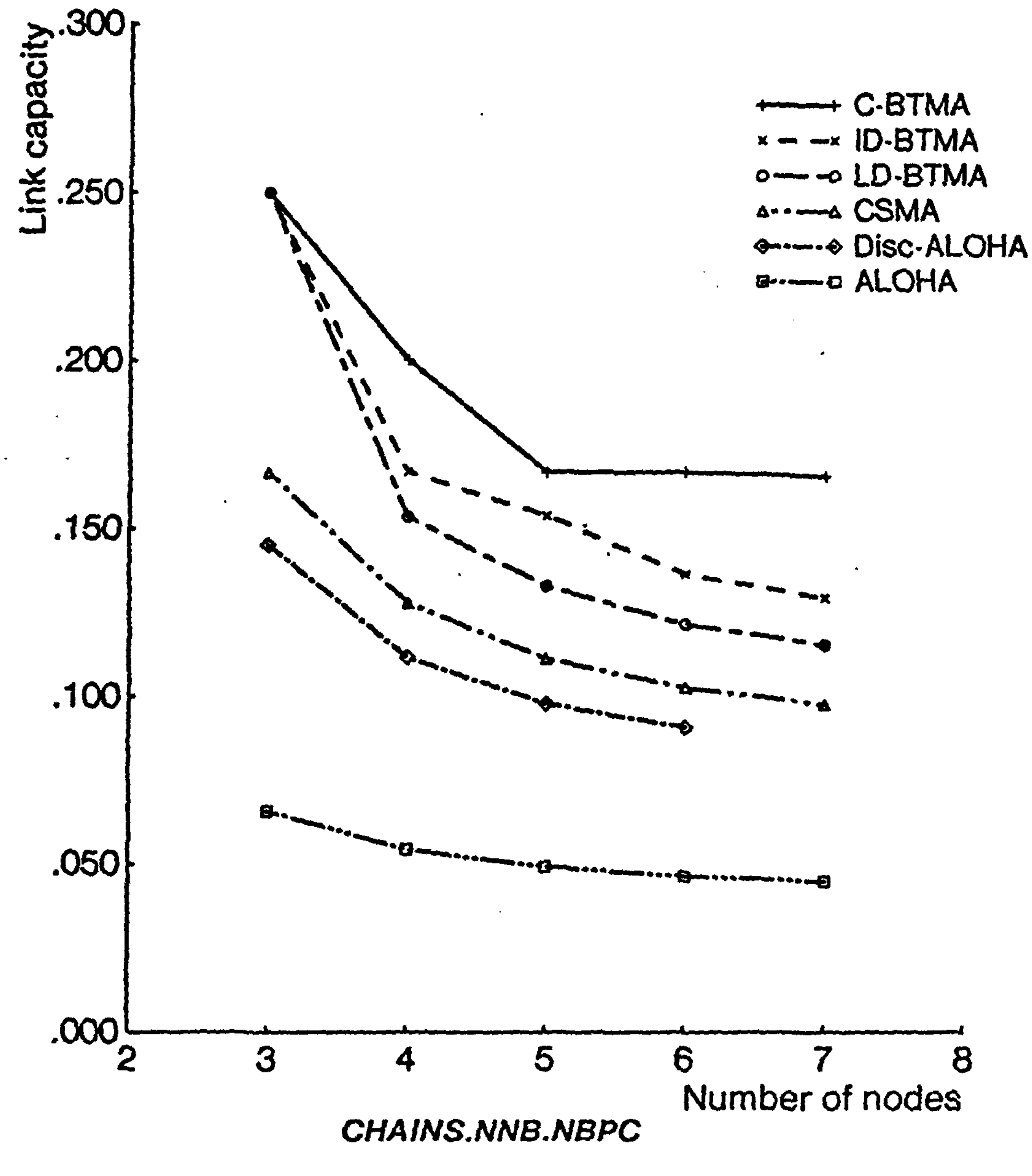


Fig. 9.15 Link capacity for chain topologies with uniform nearest-neighbor traffic, for a narrowband system with idealistic perfect capture

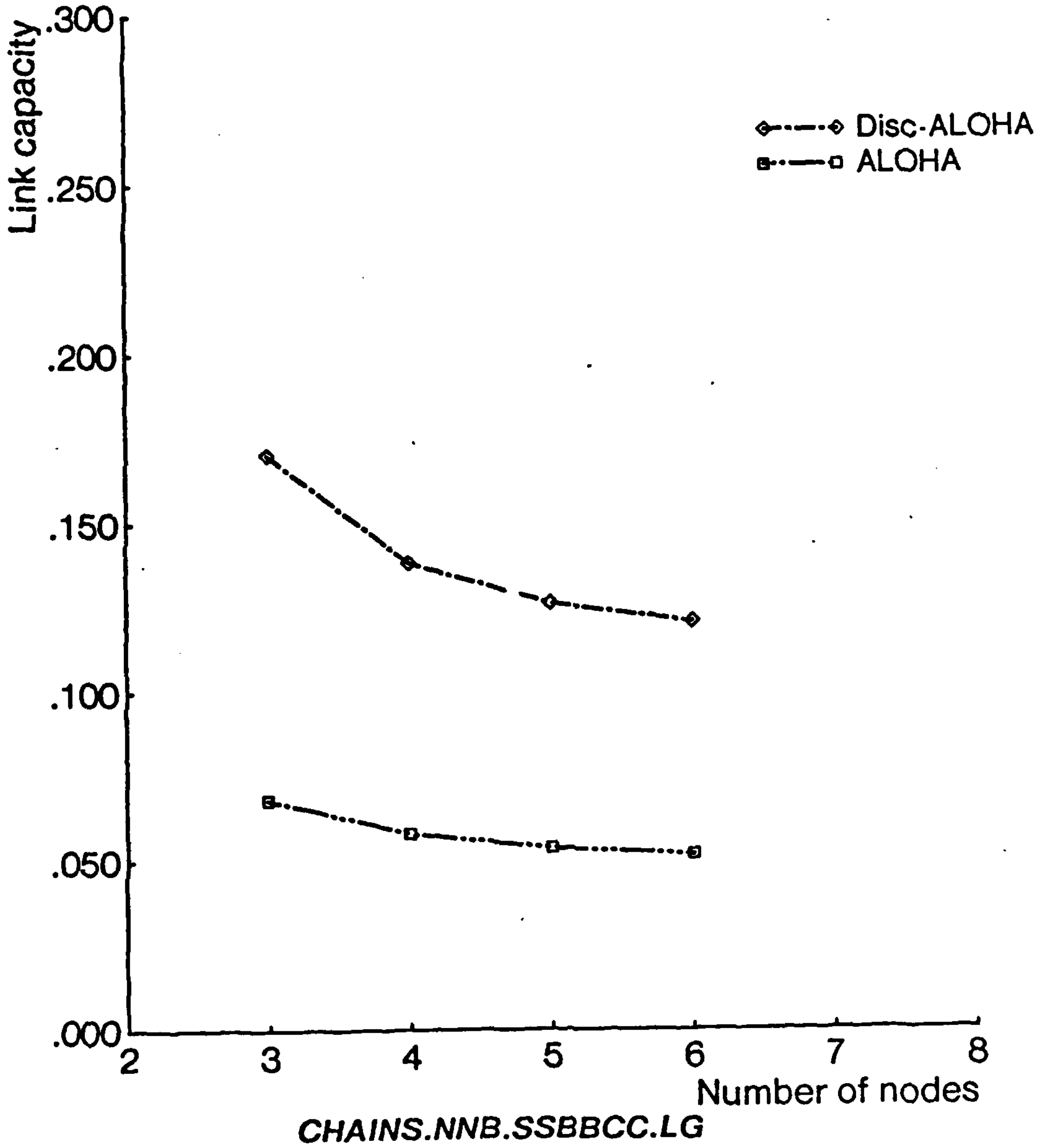


Fig. 9.16 Link capacity for chain topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and long bit duration

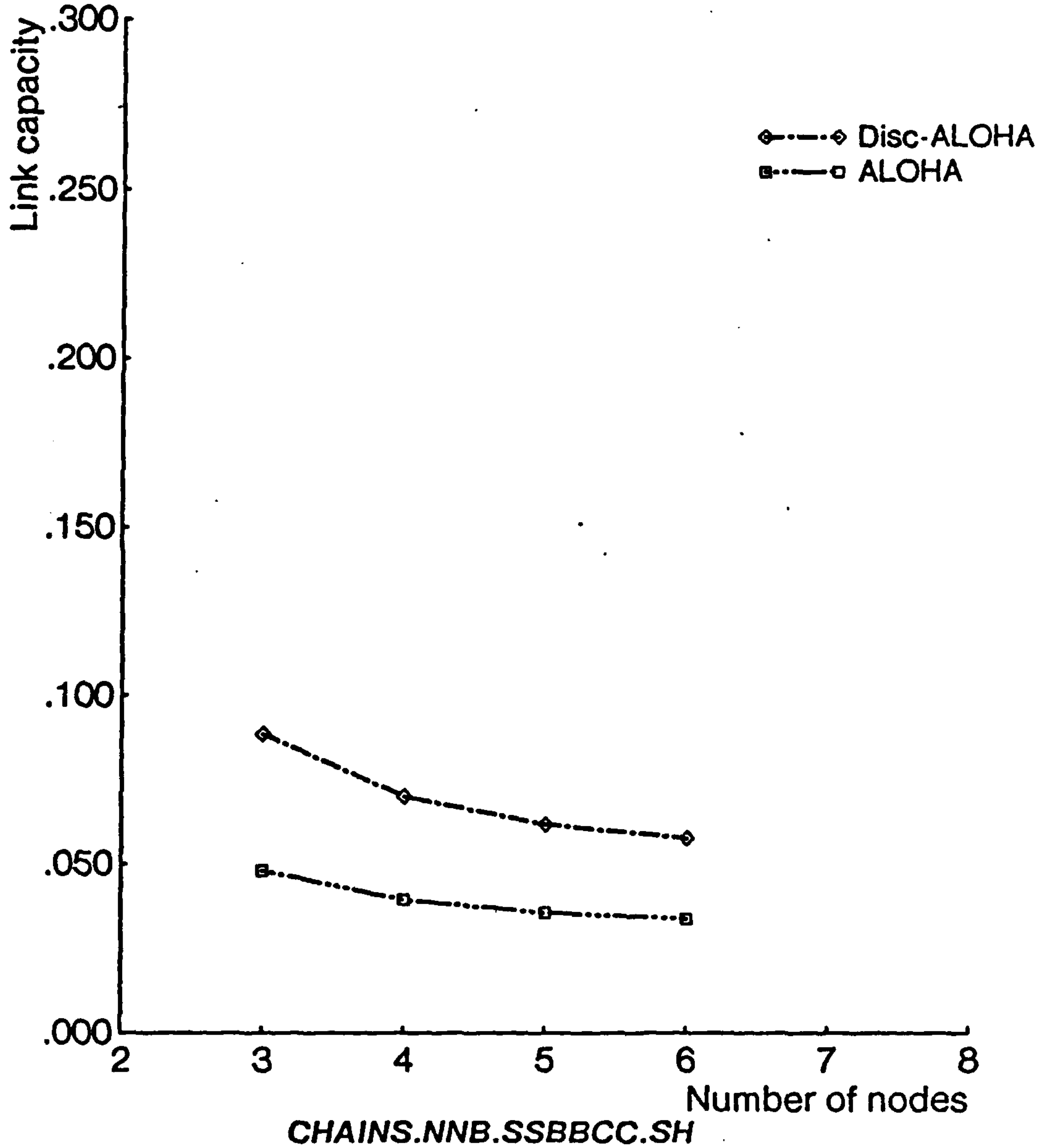


Fig. 9.17 Link capacity for chain topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and short bit duration

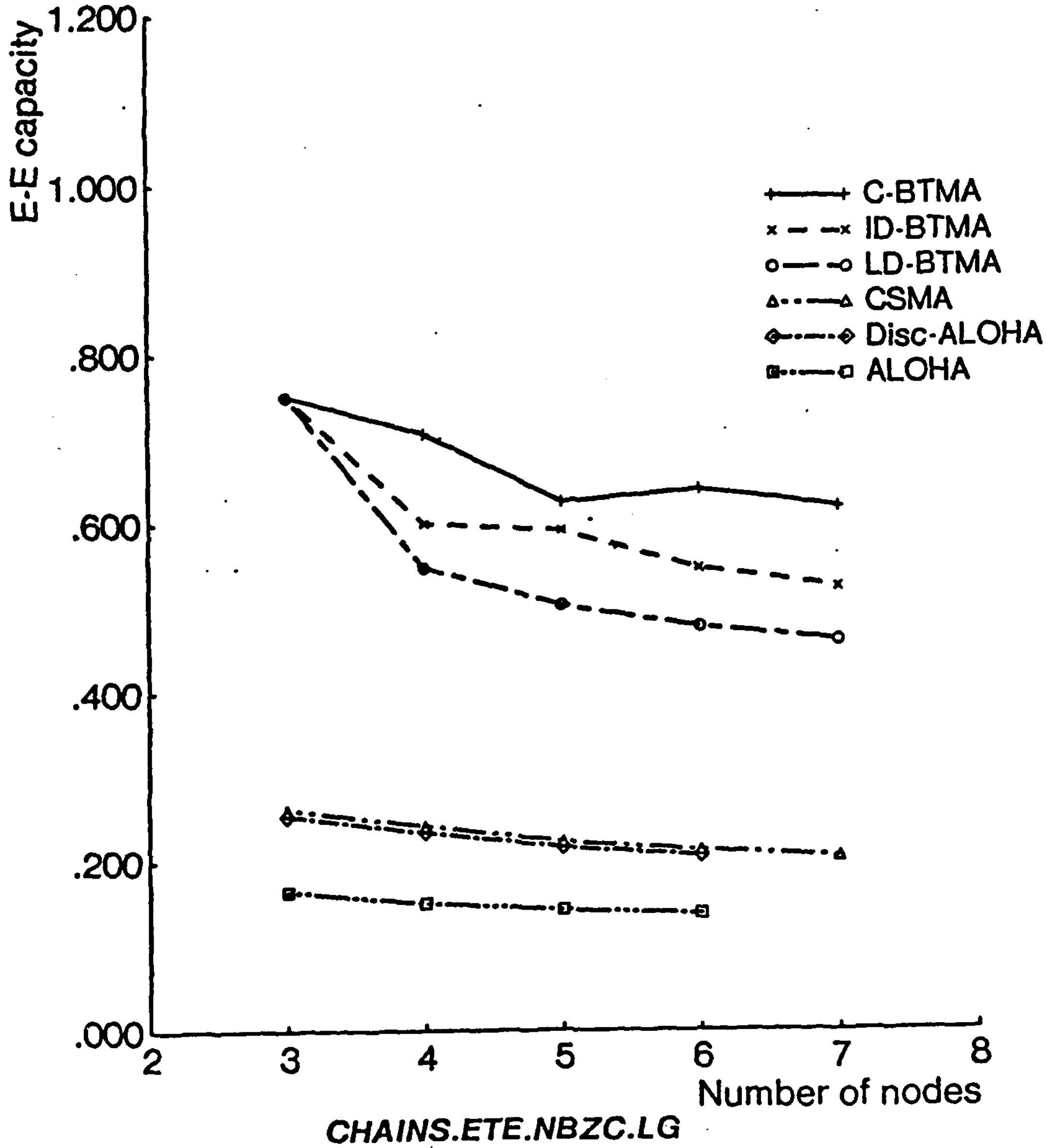


Fig. 9.18 End-to-end capacity for chain topologies with uniform end-to-end traffic, for a narrowband system with zero capture and long bit duration

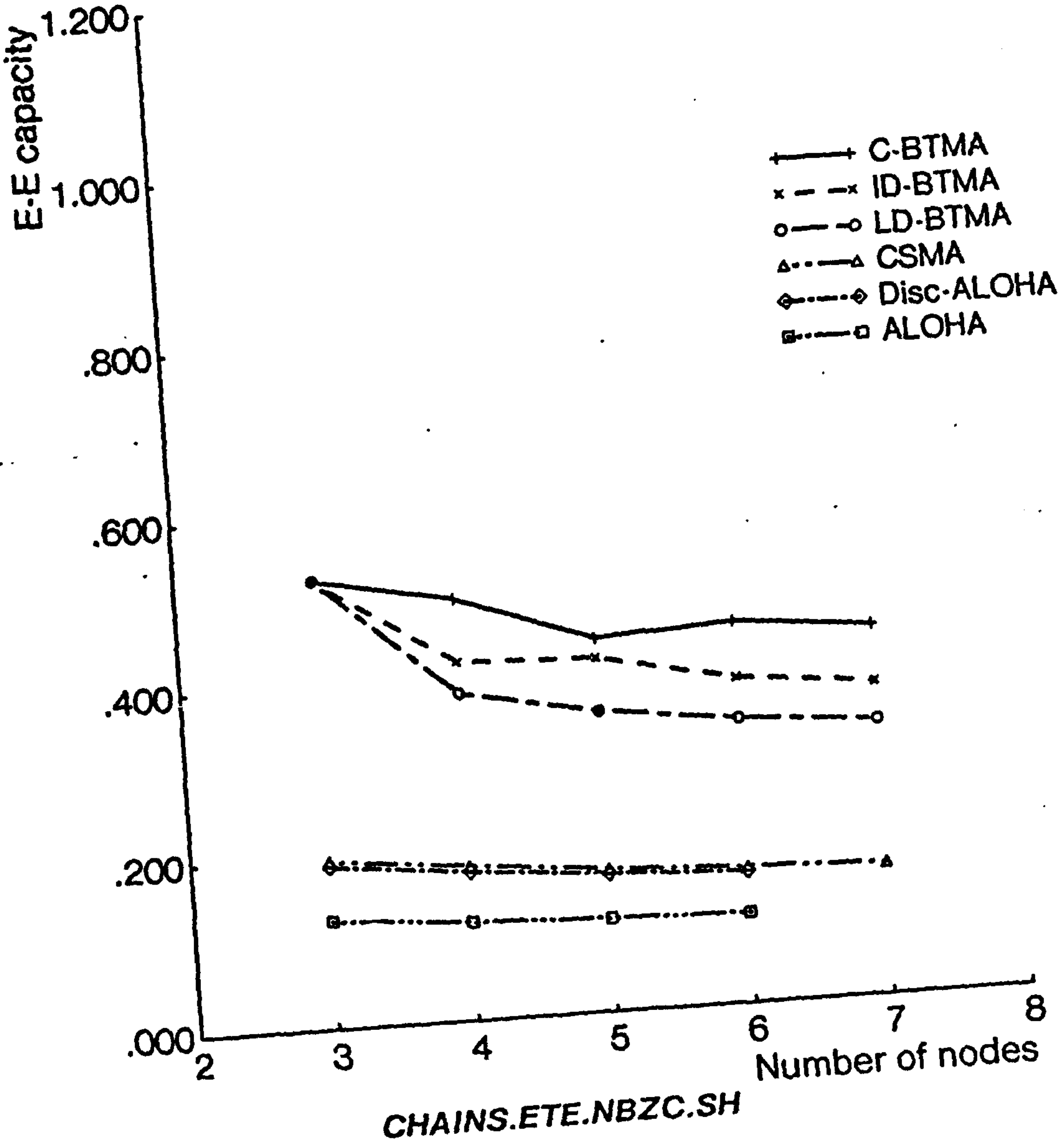


Fig. 9.19 End-to-end capacity for chain topologies with uniform end-to-end traffic, for a narrowband system with zero capture and short bit duration

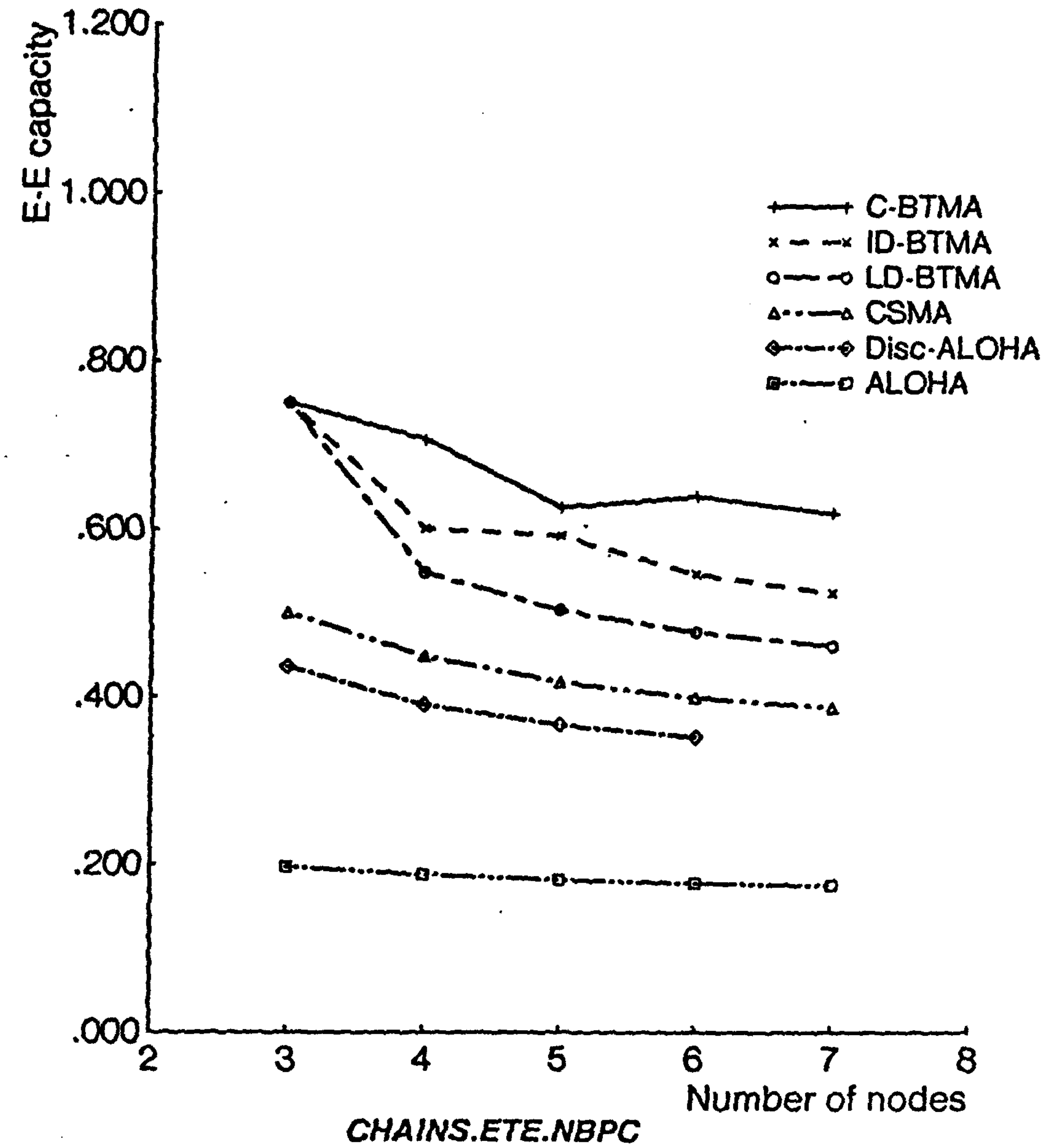


Fig. 9.20 End-to-end capacity for chain topologies with uniform end-to-end traffic, for a narrowband system with idealistic perfect capture

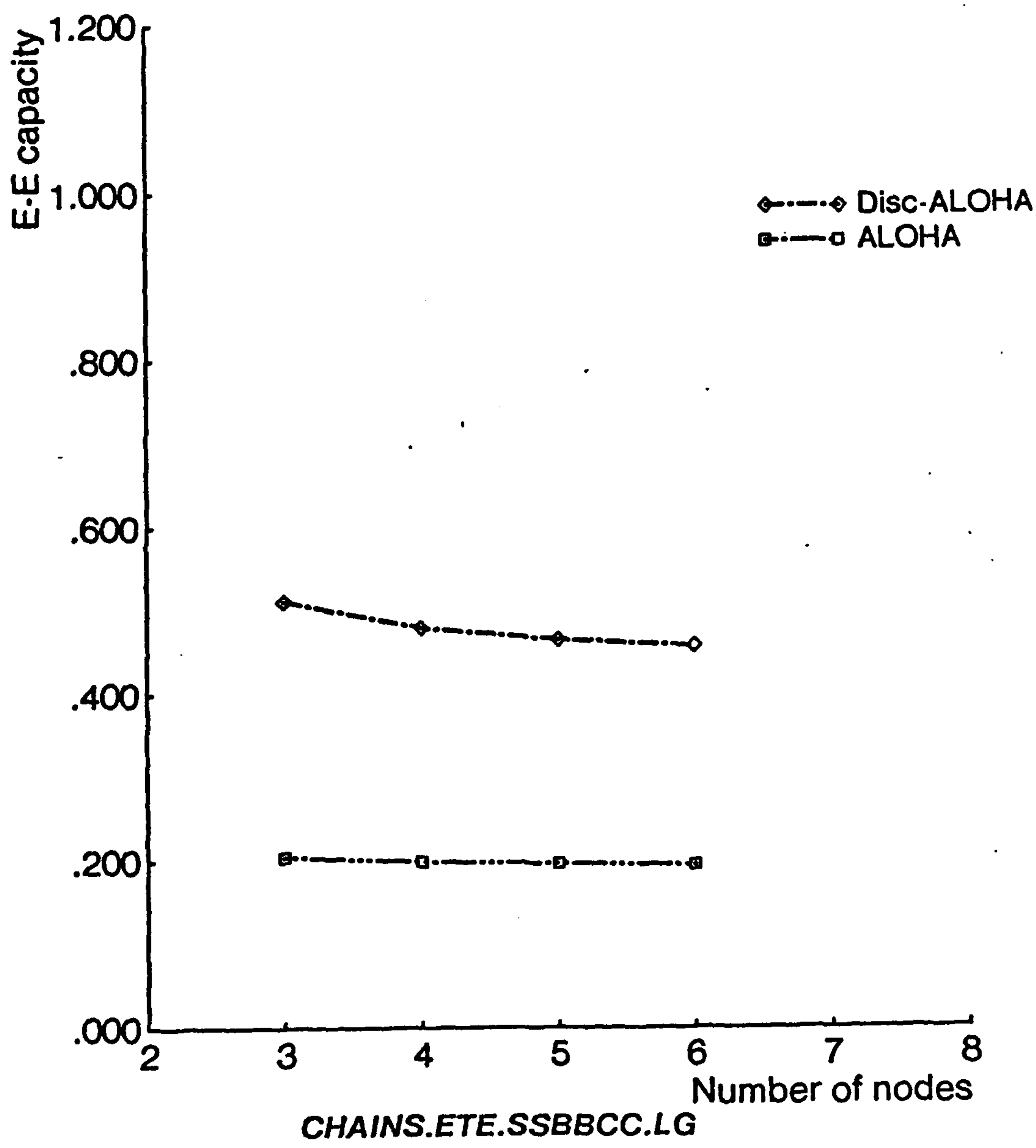


Fig. 9.21 End-to-end capacity for chain topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and long bit duration

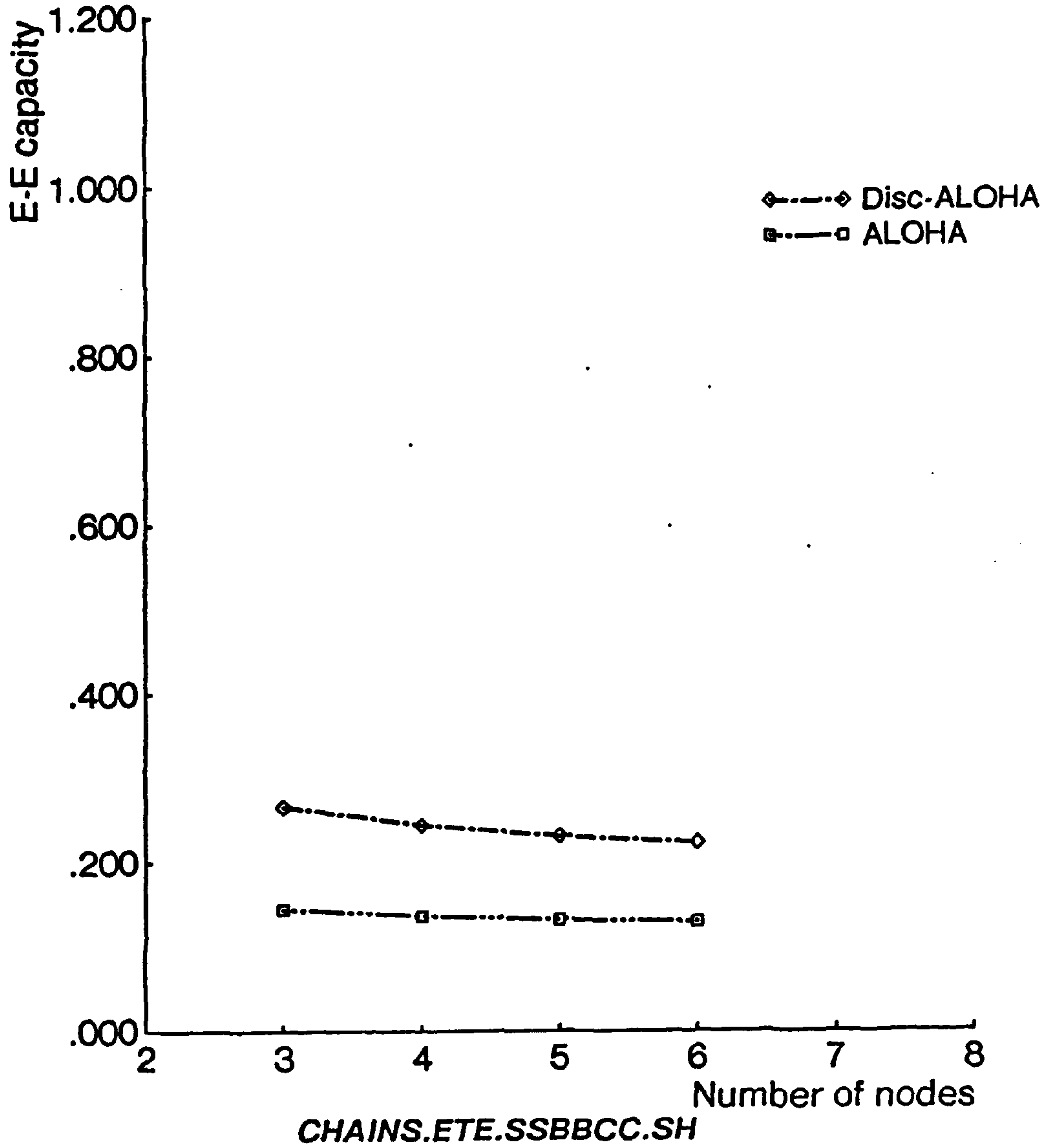


Fig. 9.22 End-to-end capacity for chain topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and short bit duration

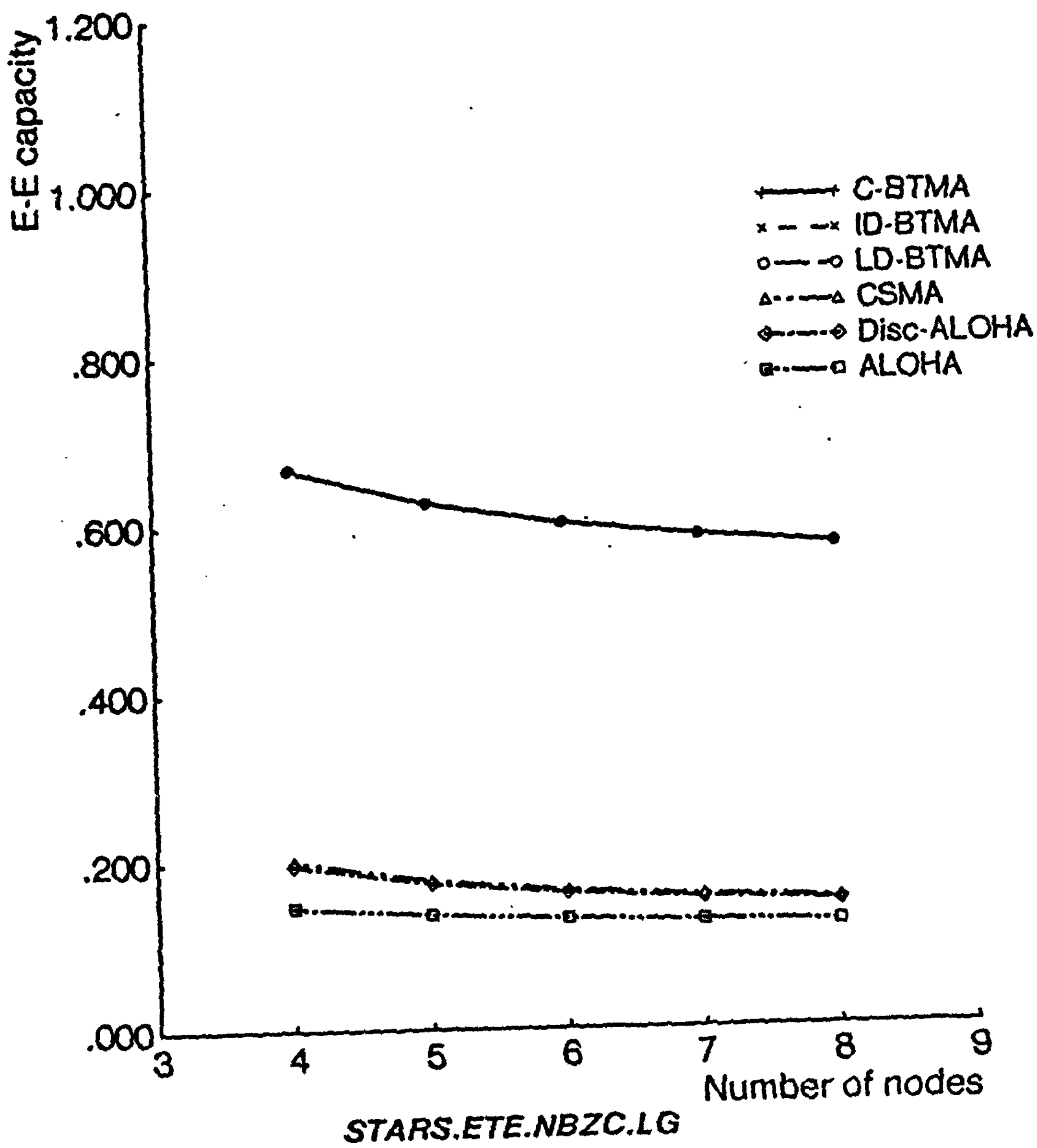


Fig. 9.23 End-to-end capacity for star topologies with uniform end-to-end traffic, for a narrowband system with zero capture and long bit duration

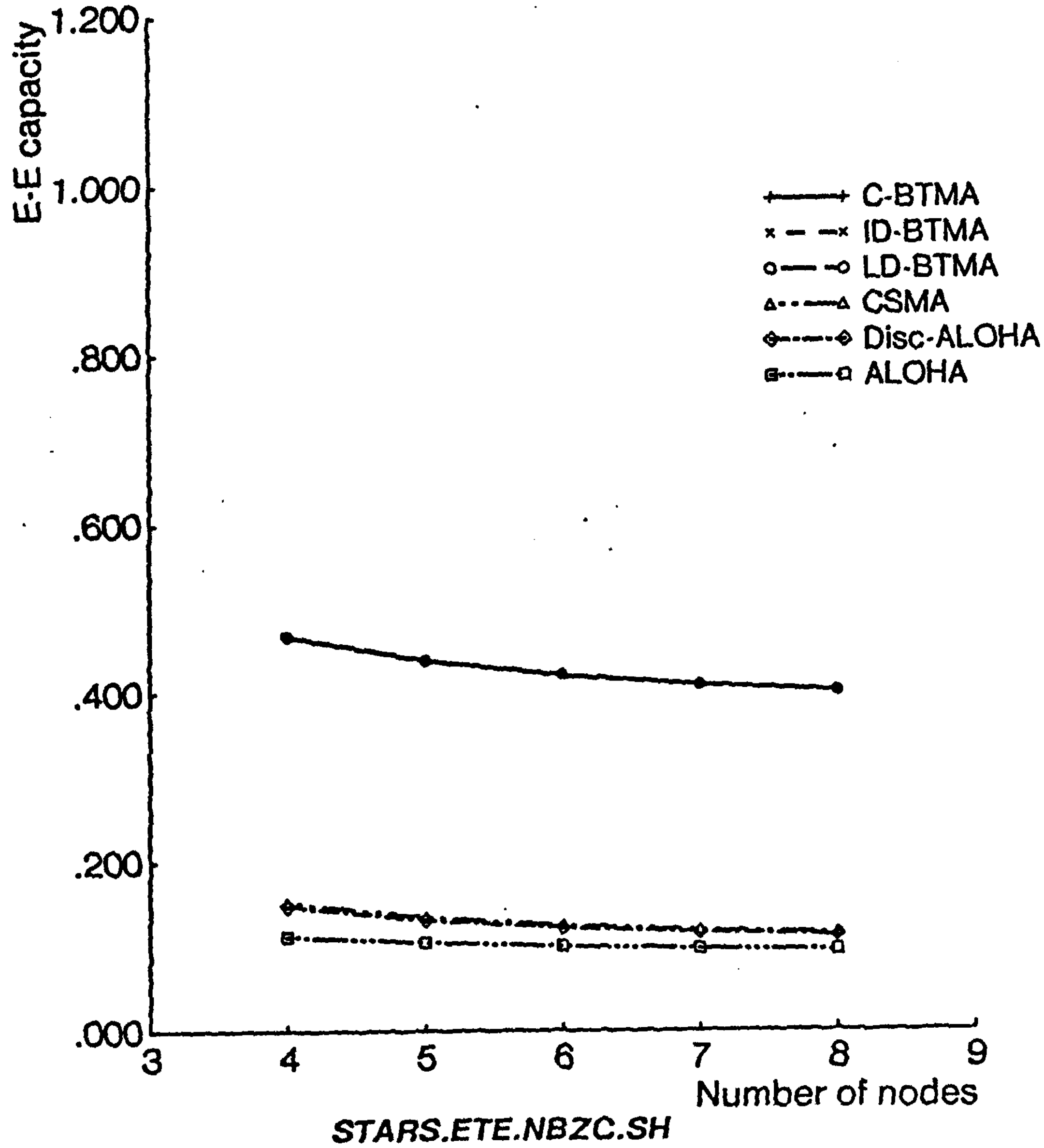


Fig. 9.24 End-to-end capacity for star topologies with uniform end-to-end traffic, for a narrowband system with zero capture and short bit duration

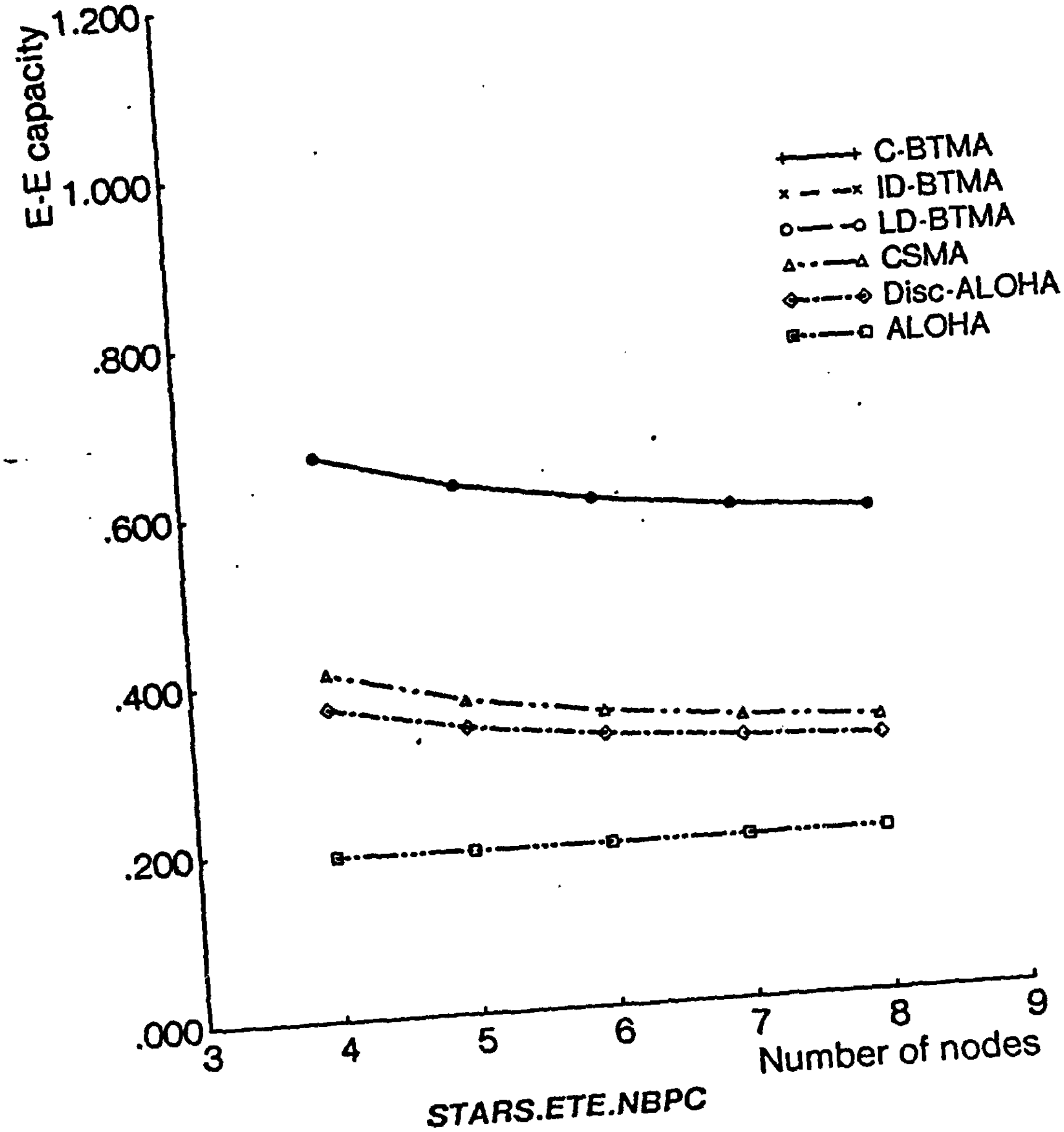


Fig. 9.25 End-to-end capacity for star topologies with uniform end-to-end traffic, for a narrowband system with idealistic perfect capture

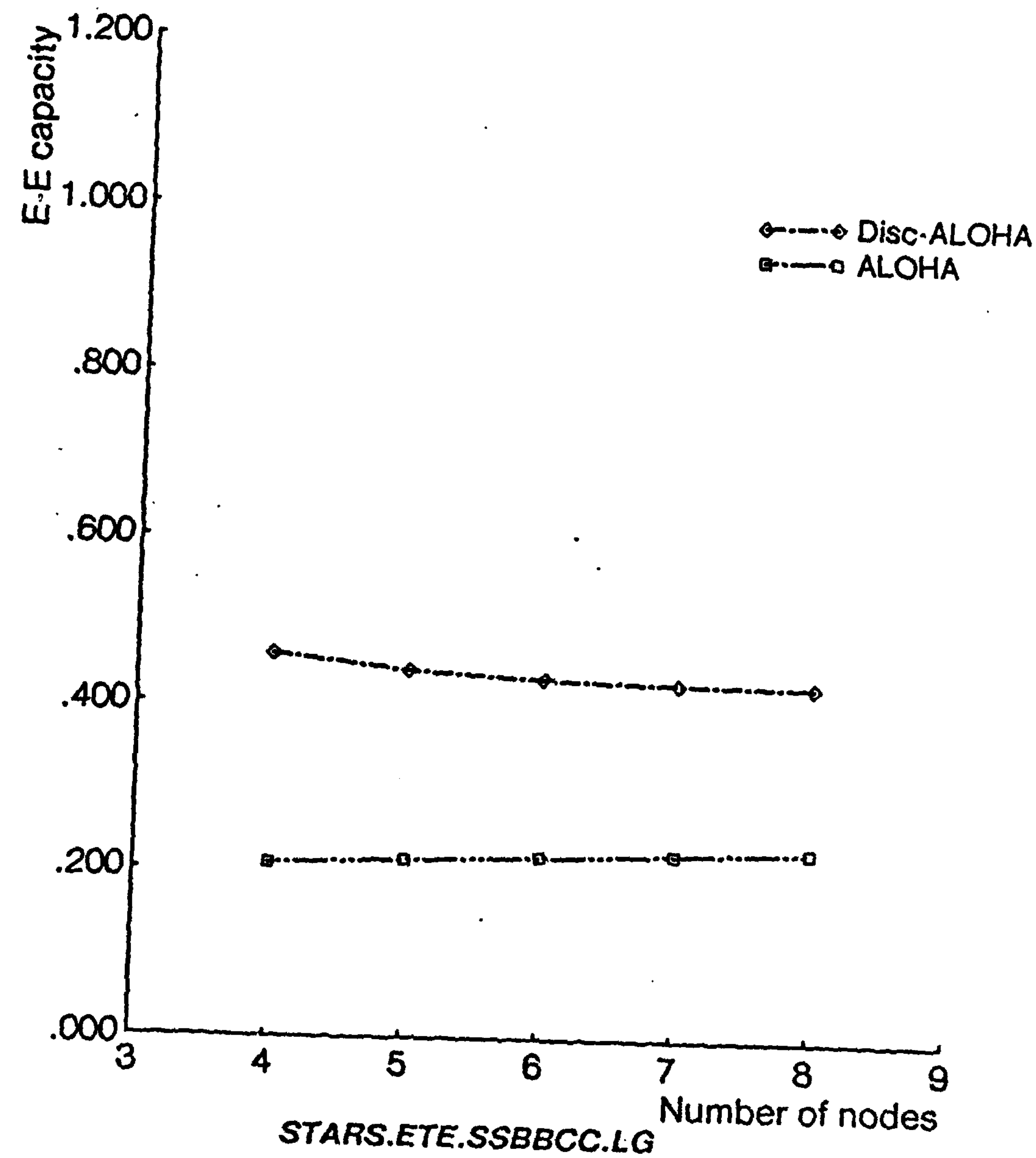


Fig. 9.26 End-to-end capacity for star topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and long bit duration

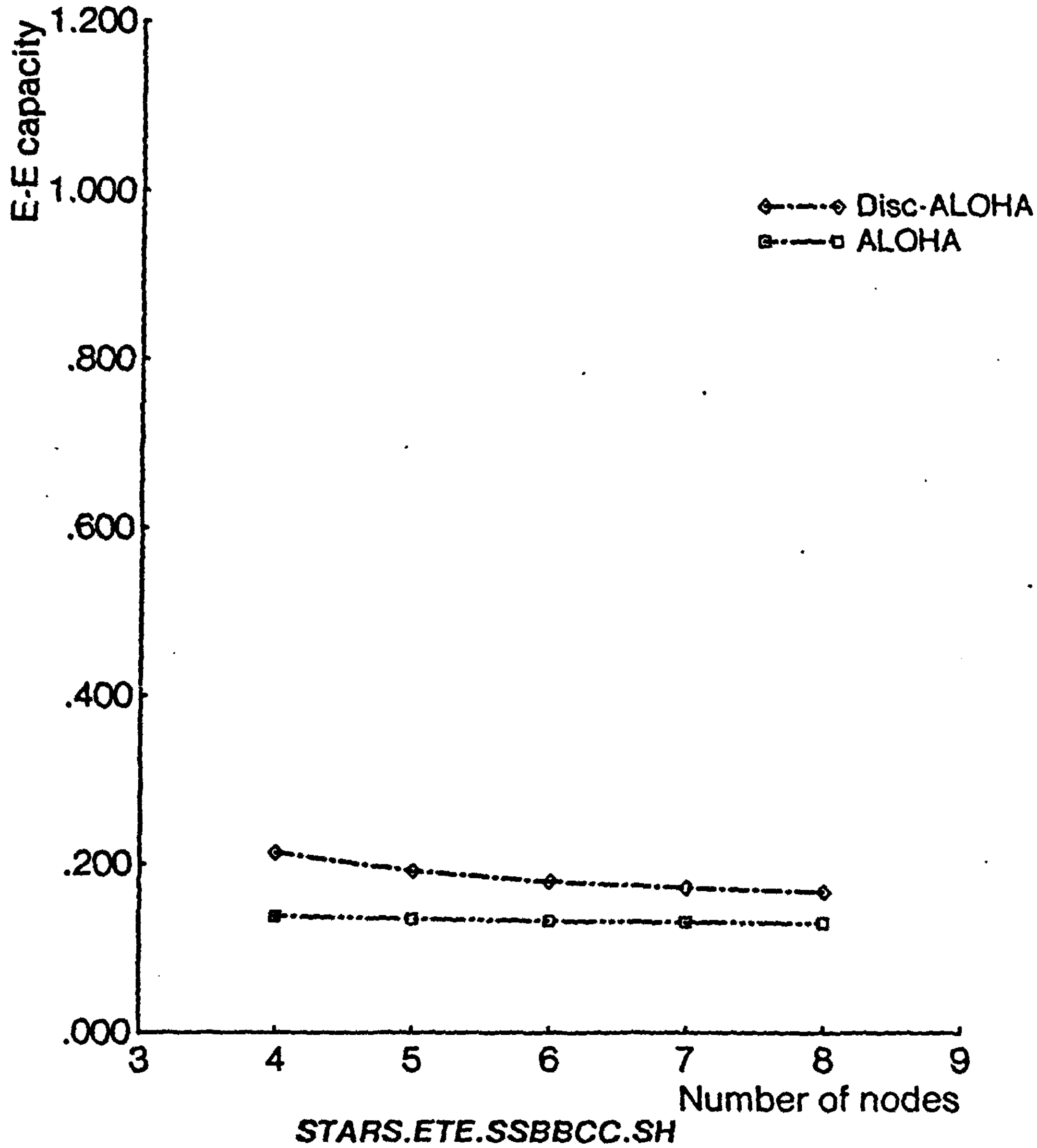


Fig. 9.27 End-to-end capacity for star topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and short bit duration

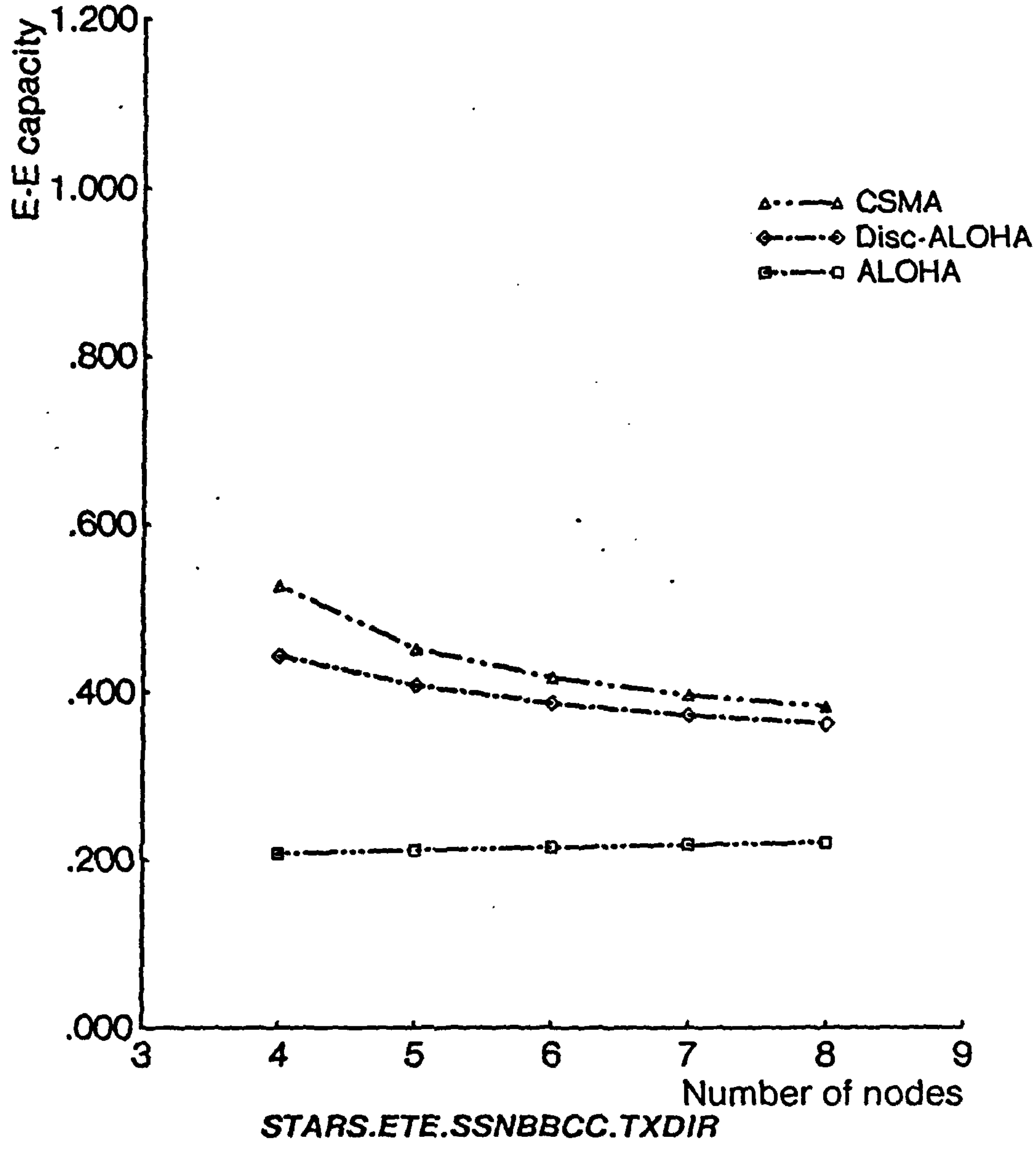


Fig. 9.28 End-to-end capacity for star topologies with uniform *end-to-end* traffic, for a spread spectrum system with bit-homogeneous transmitter-directed codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

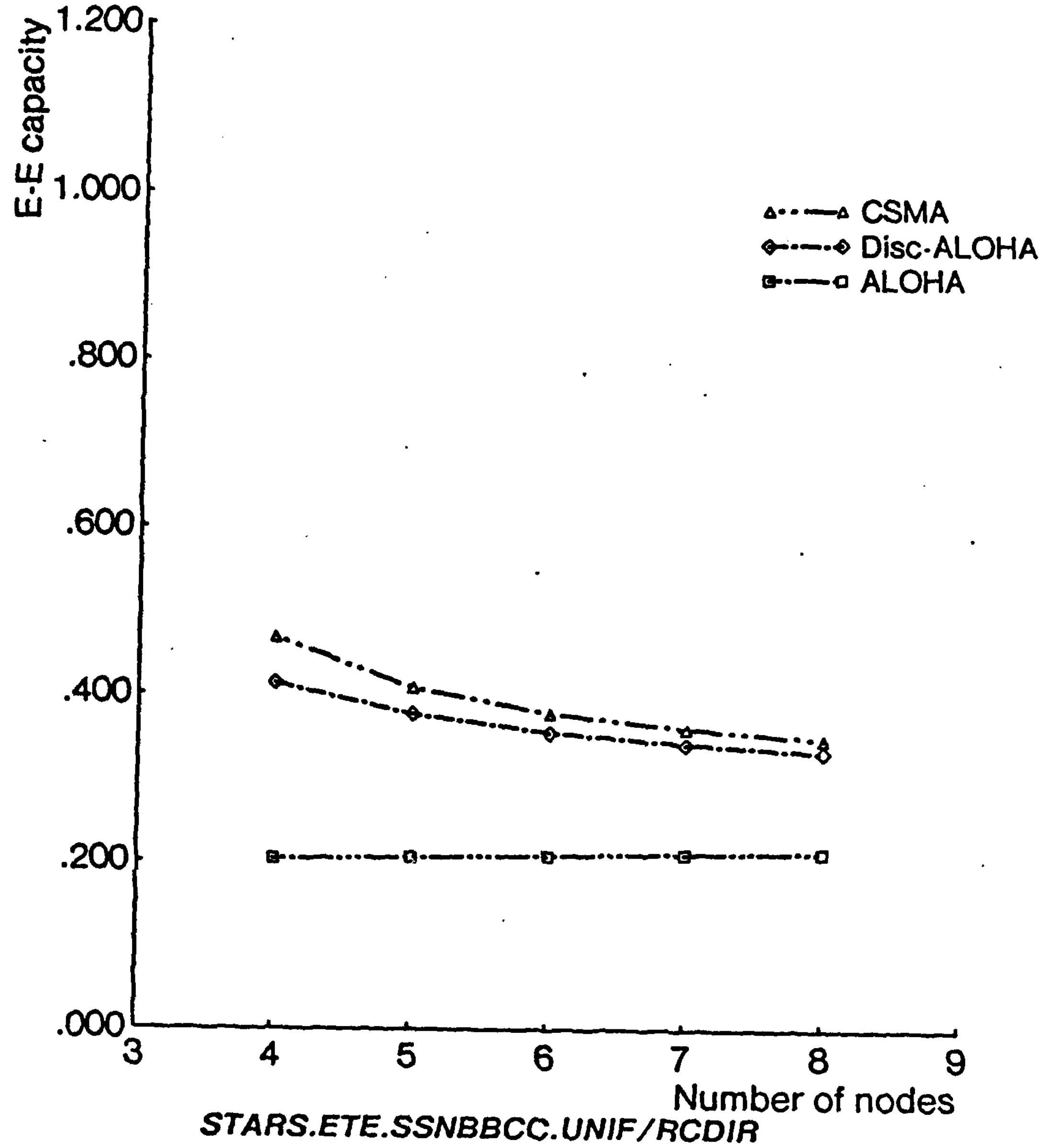


Fig. 9.29 End-to-end capacity for star topologies with uniform end-to-end traffic, for a spread spectrum system with bit-homogeneous receiver-directed or uniform codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

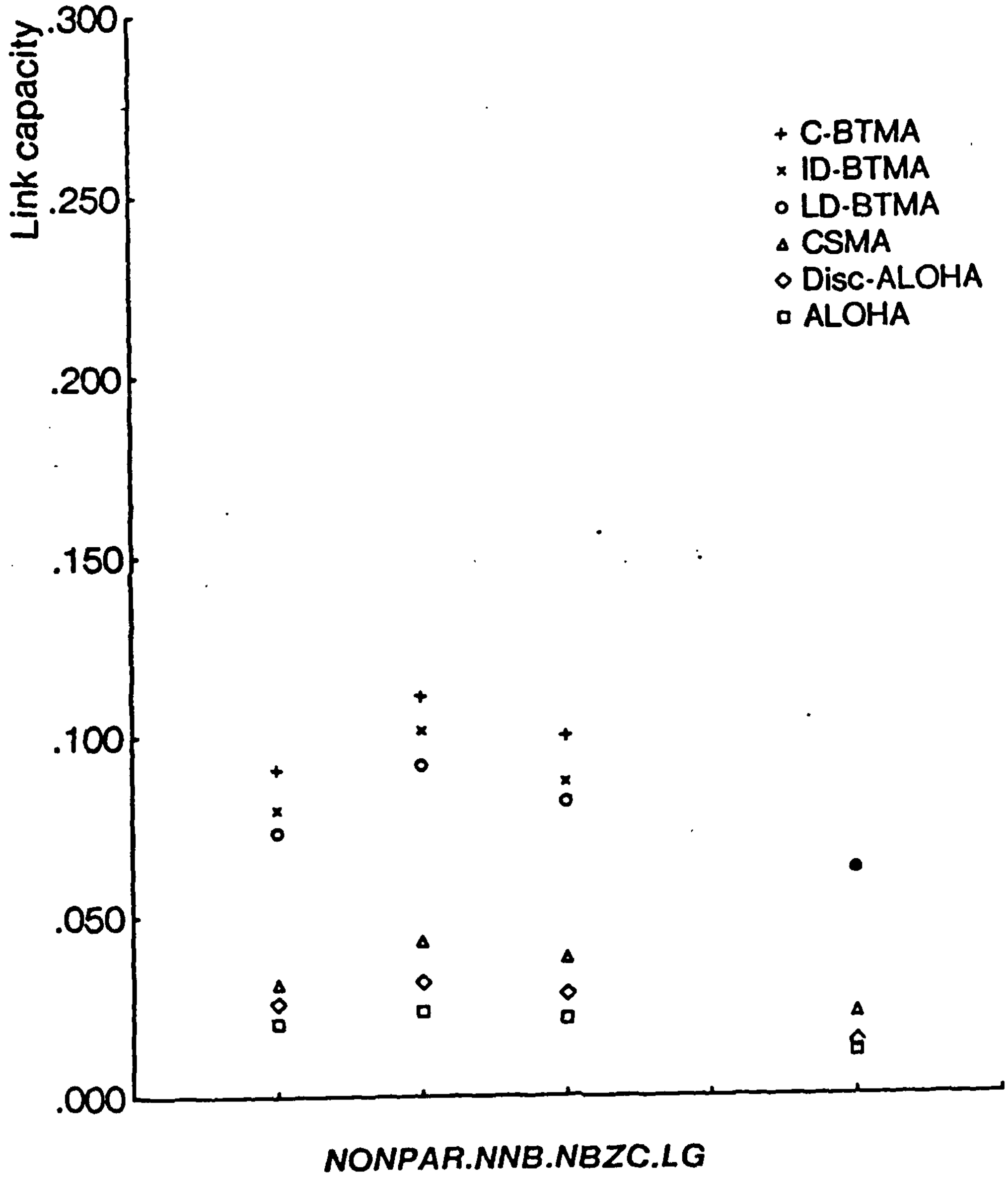


Fig. 9.30 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and long bit duration

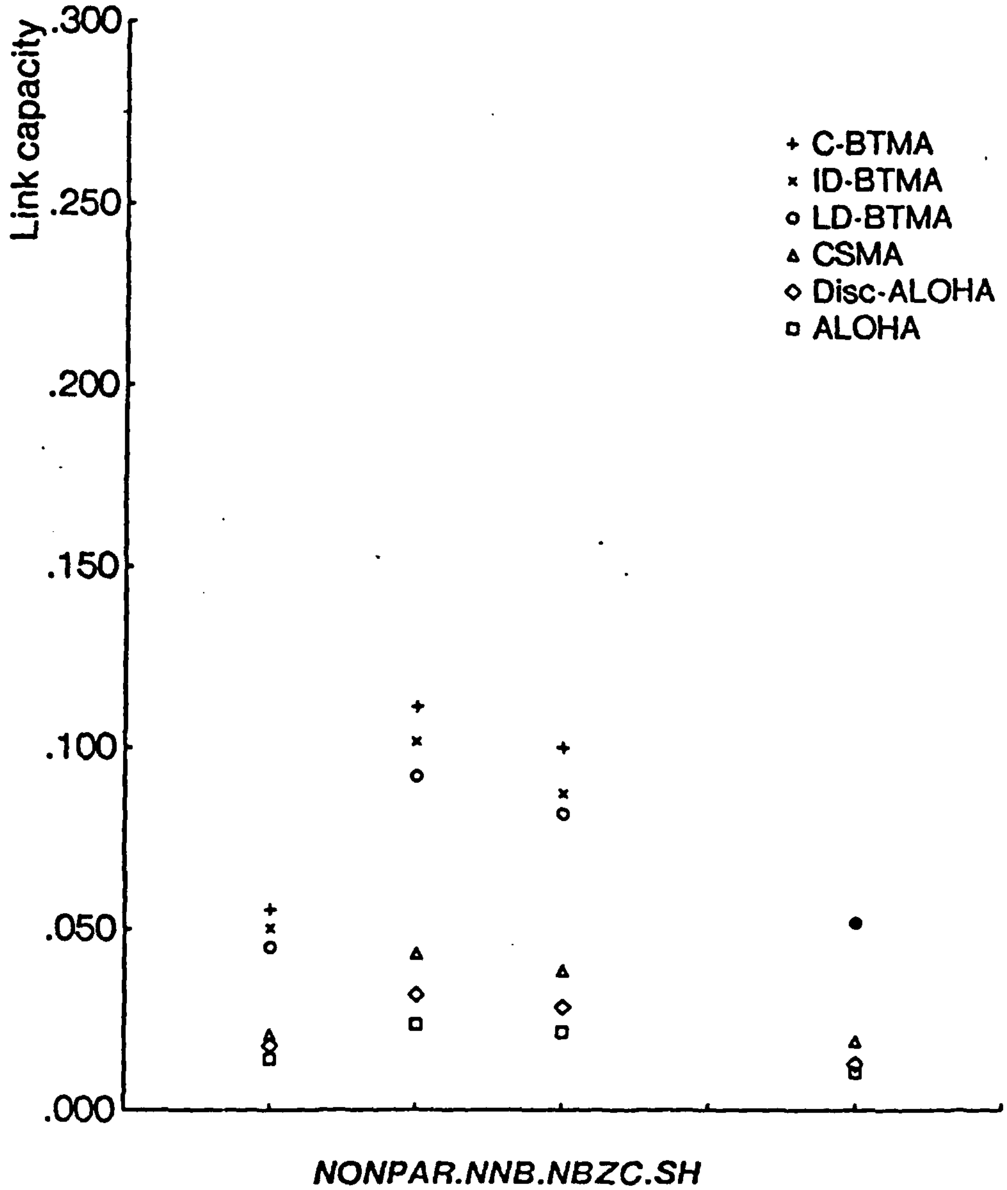


Fig. 9.31 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a narrowband system with zero capture and short bit duration

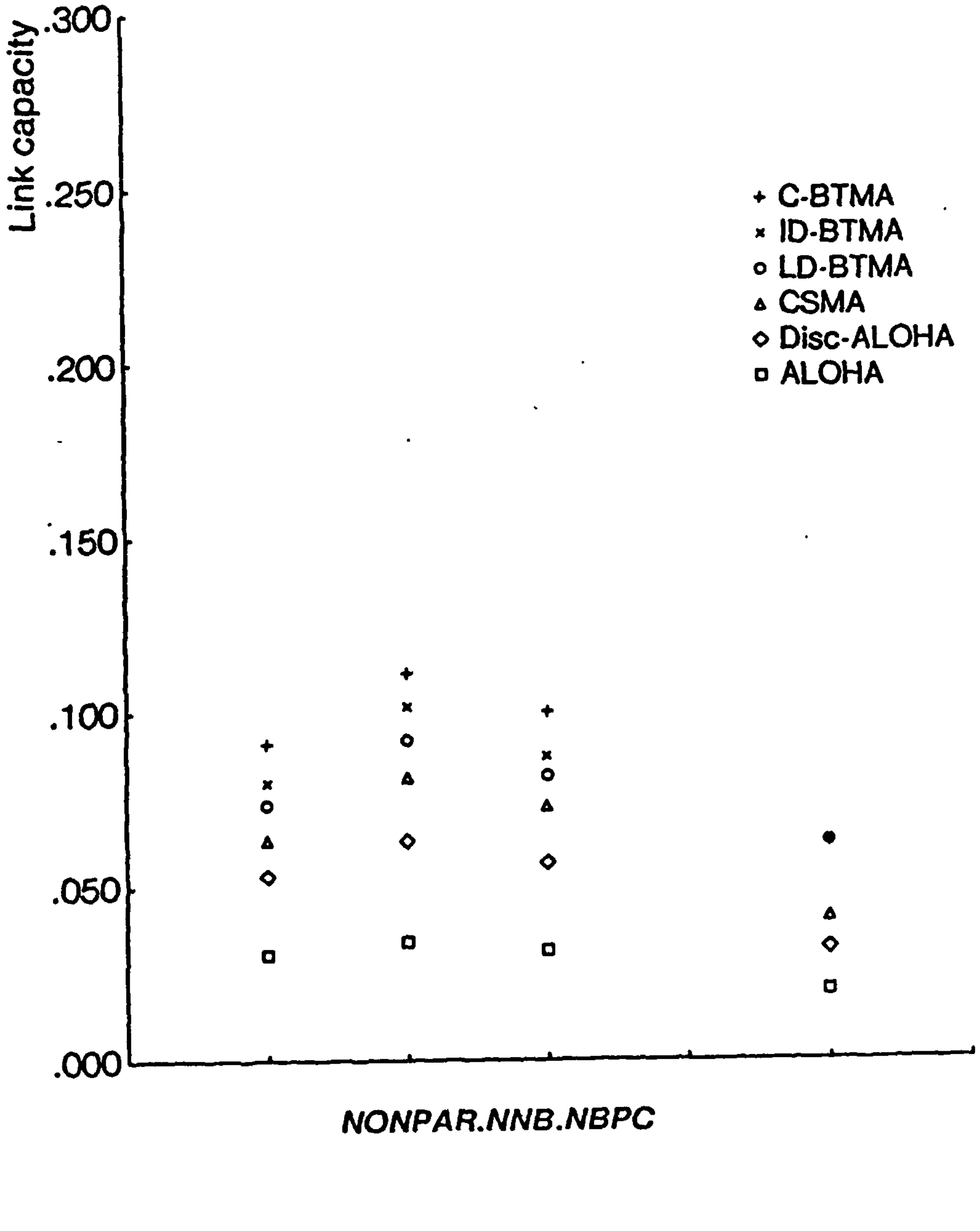


Fig. 9.32 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a narrowband system with idealistic perfect capture

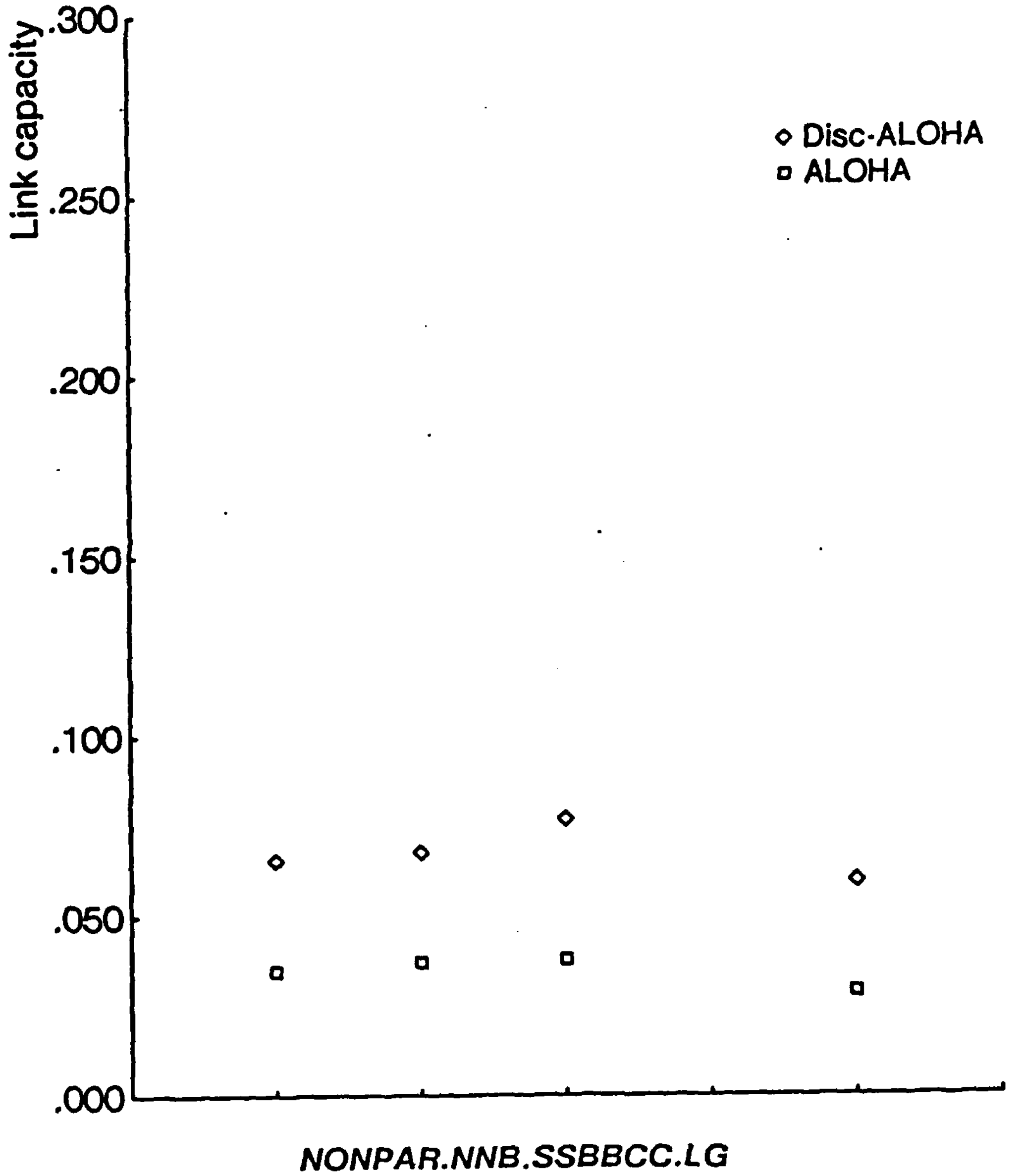


Fig. 9.33 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and long bit duration

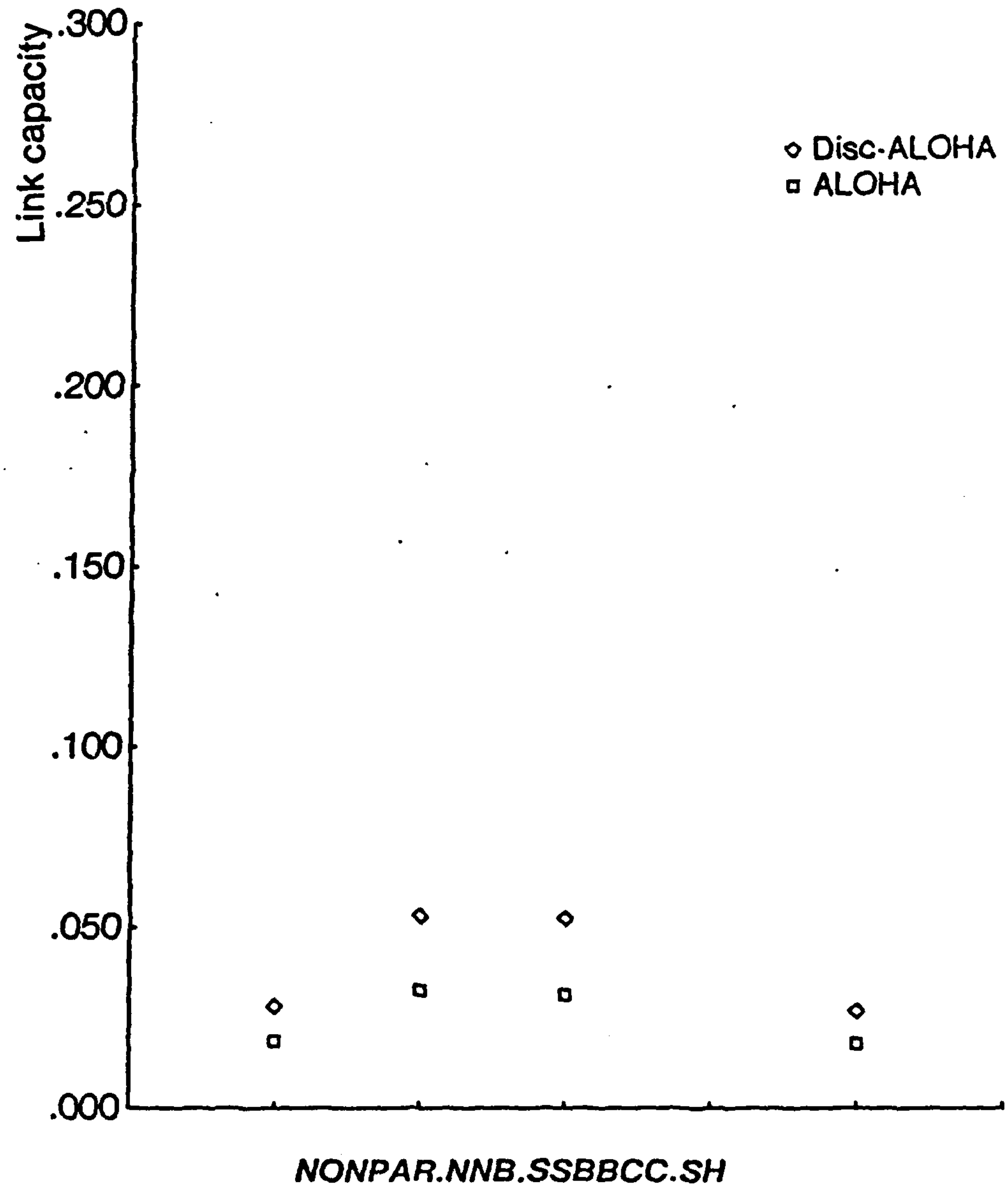


Fig. 9.34 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-changing codes and short bit duration

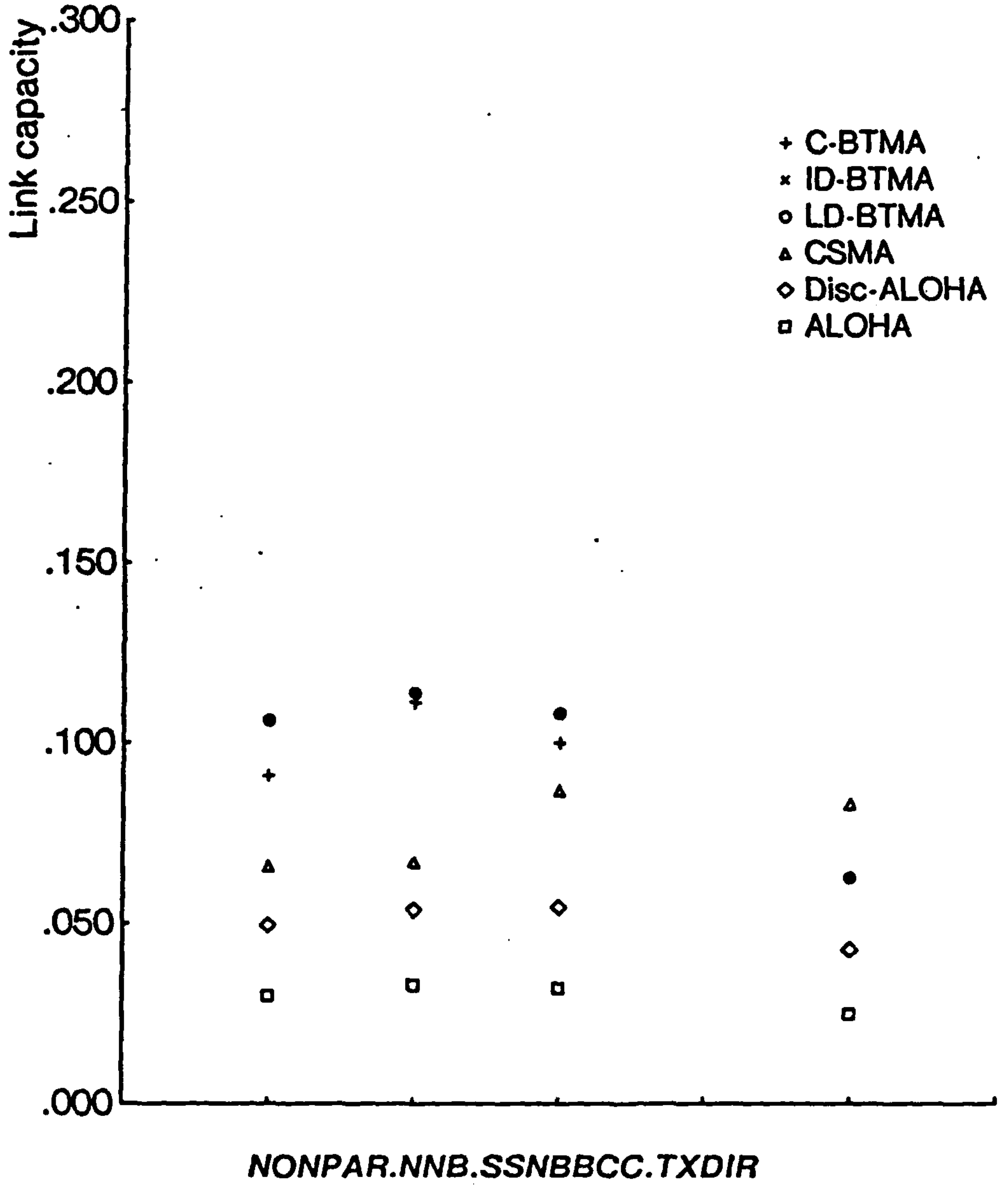


Fig. 9.35 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous transmitter-directed codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

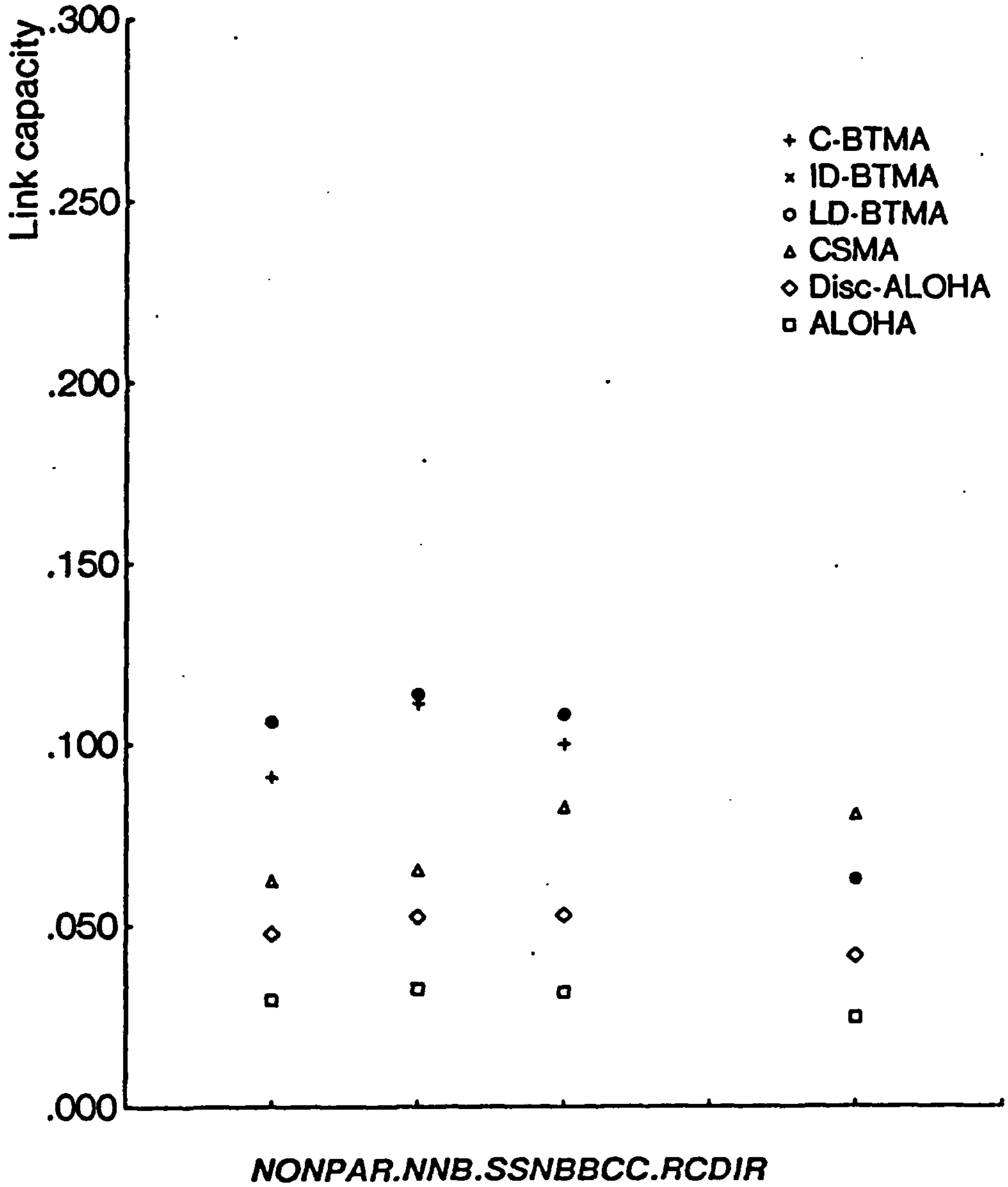


Fig. 9.36 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous receiver-directed codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

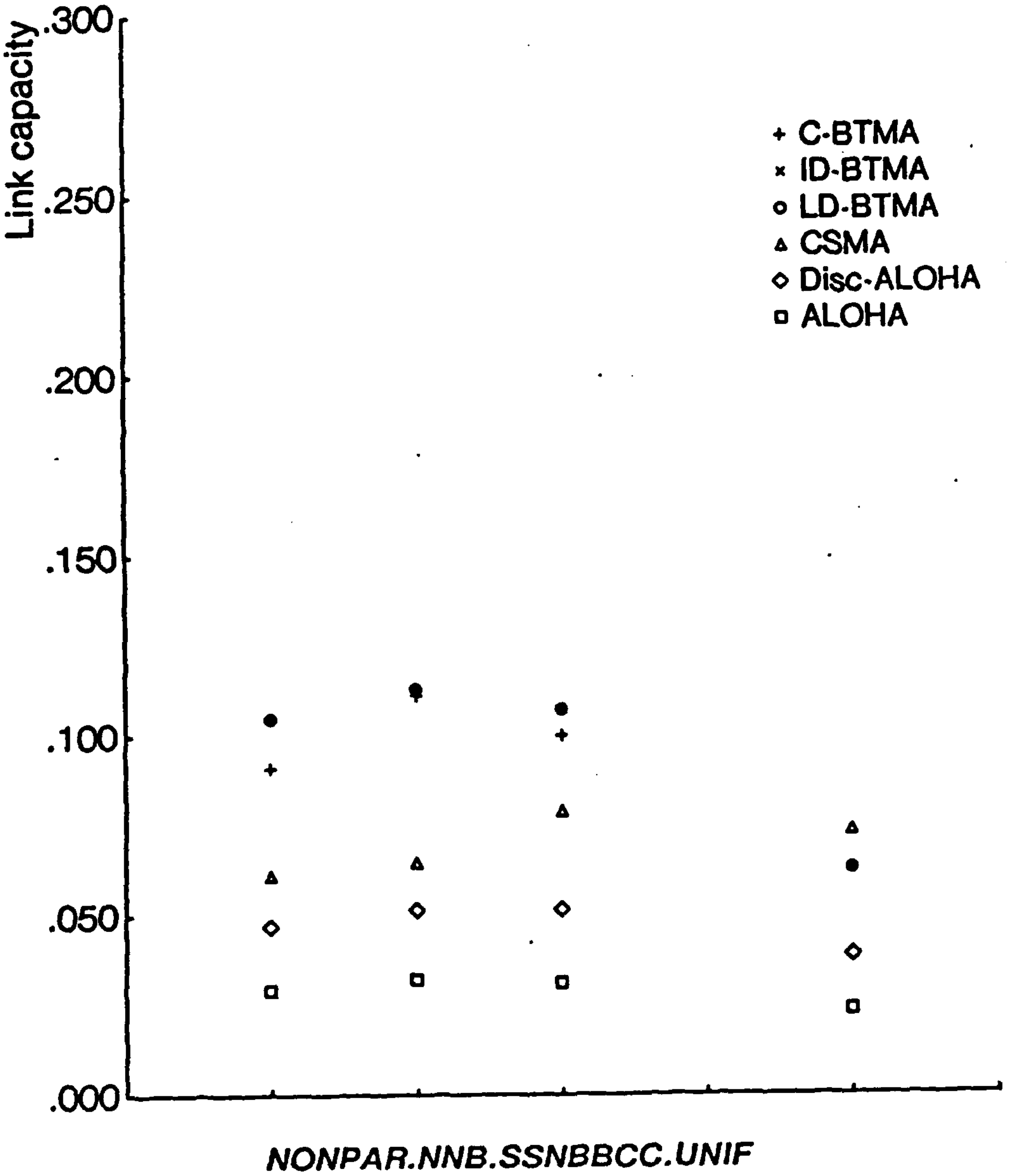


Fig. 9.37 Link capacity for nonparametric topologies with uniform nearest-neighbor traffic, for a spread spectrum system with bit-homogeneous uniform codes, short bit duration, and a fourfold increase in transmitted power (see text for details)

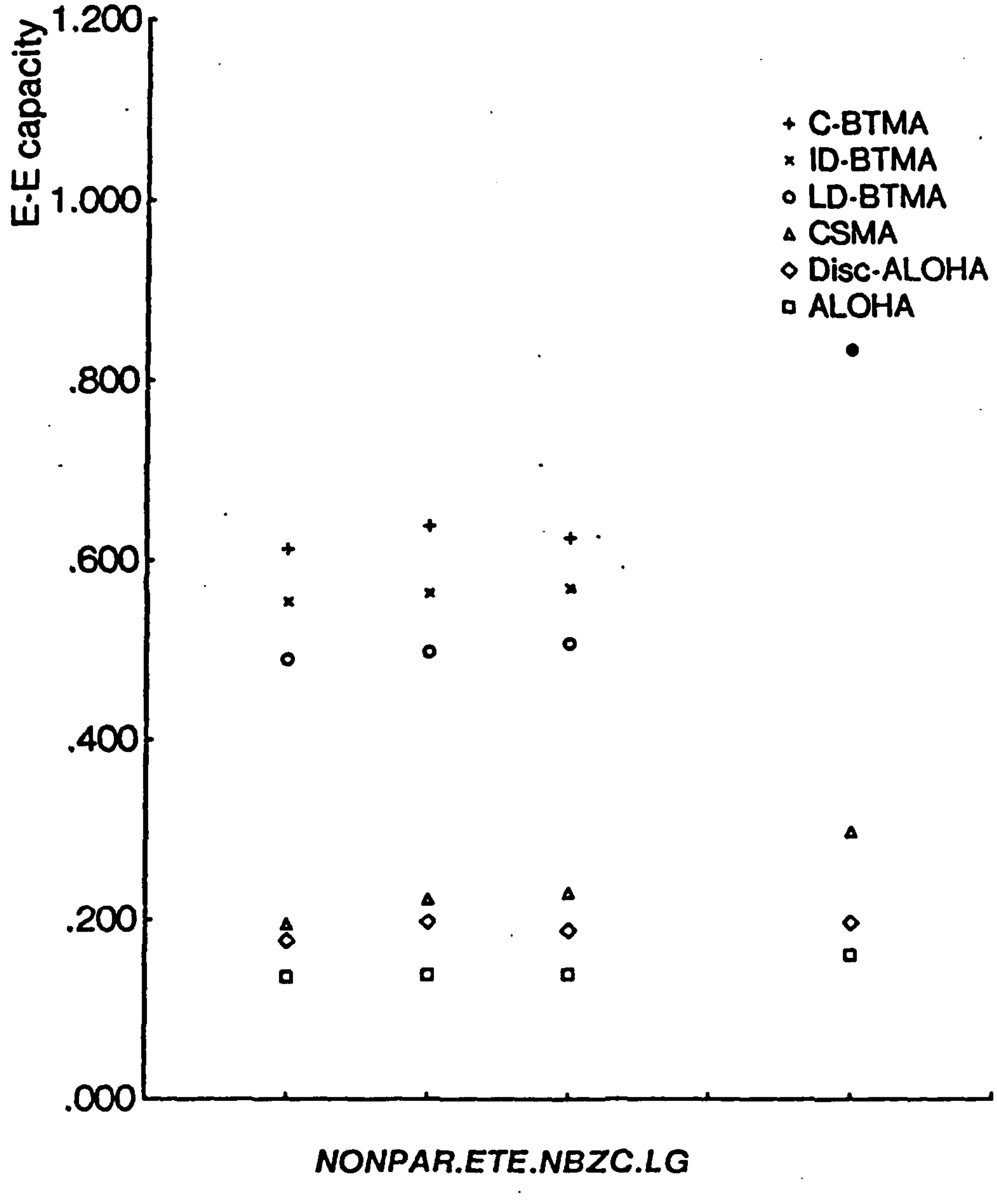


Fig. 9.38 End-to-end capacity for nonparametric topologies with uniform end-to-end traffic, for a narrowband system with zero capture and long bit duration

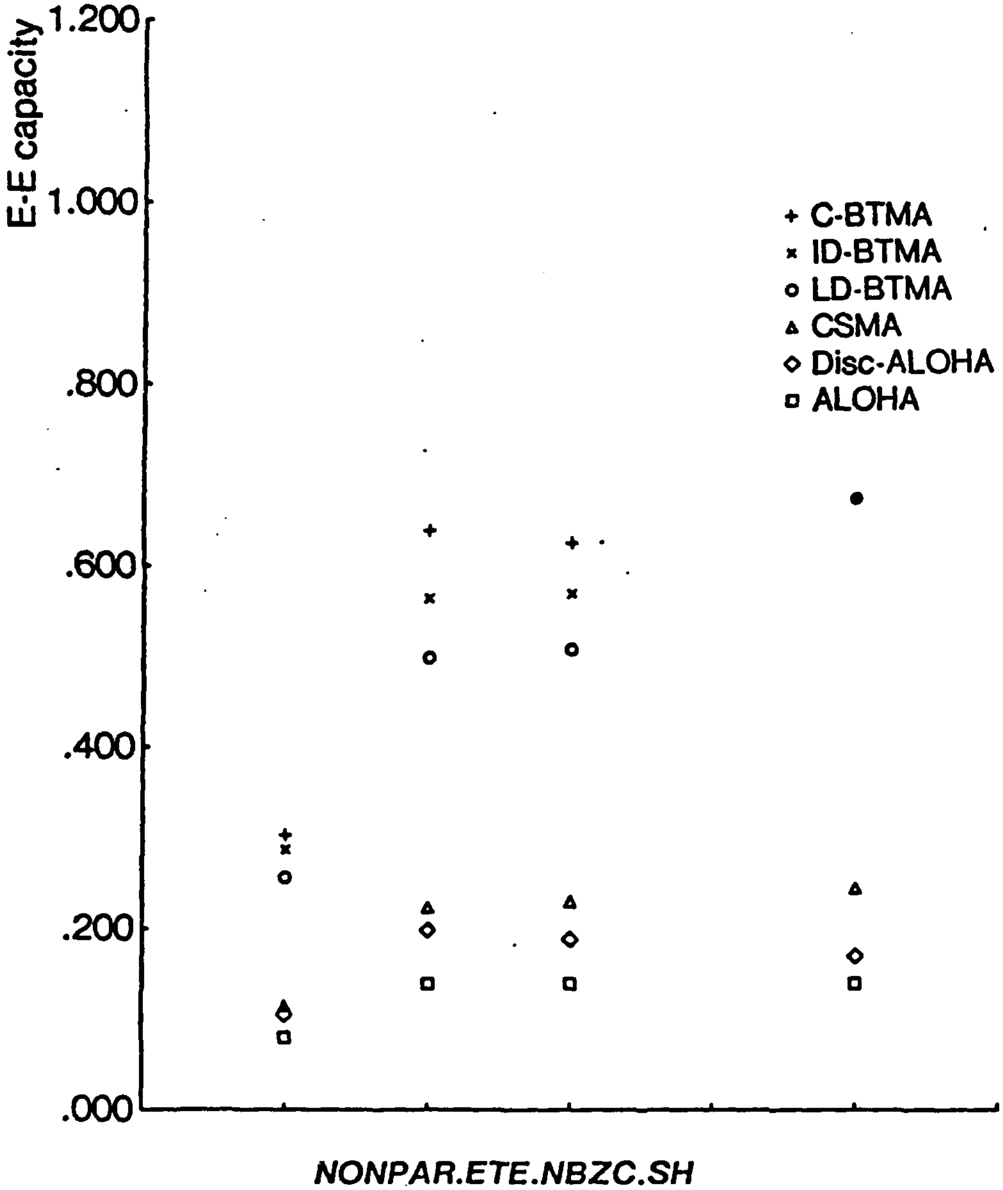


Fig. 9.39 End-to-end capacity for nonparametric topologies with uniform end-to-end traffic, for a narrowband system with zero capture and short bit duration

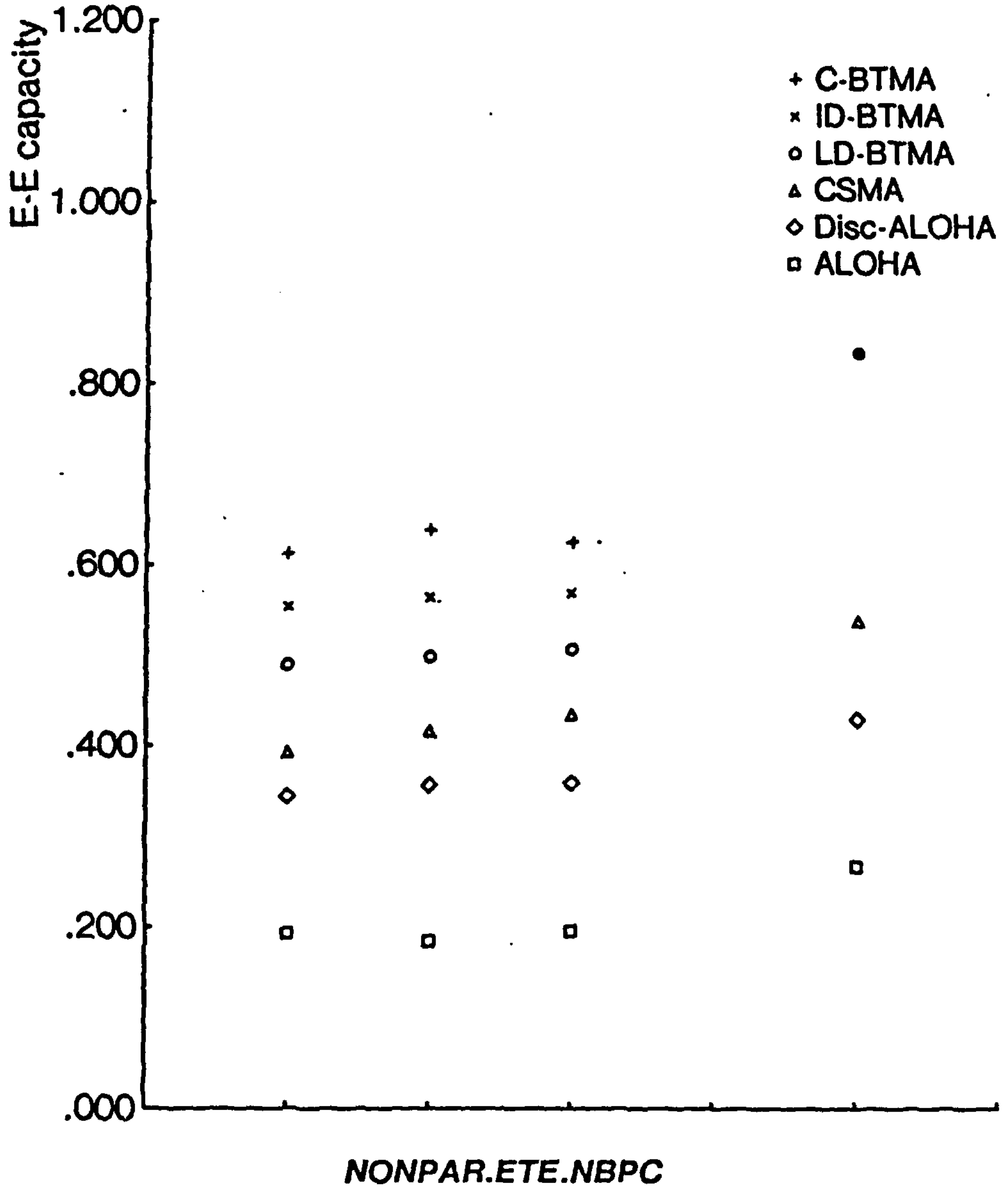


Fig. 9.40 End-to-end capacity for nonparametric topologies with uniform end-to-end traffic, for a narrowband system with idealistic perfect capture

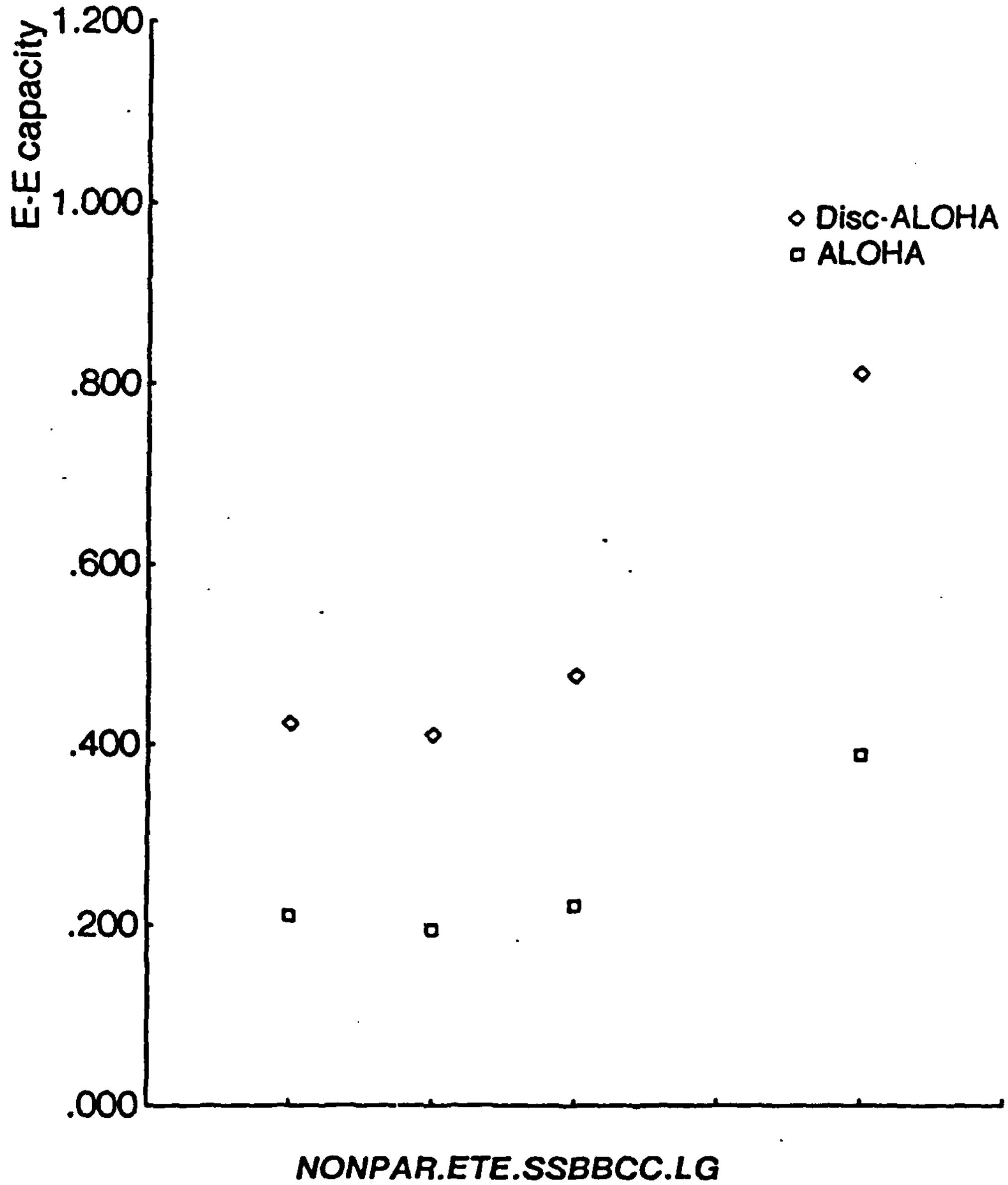


Fig. 9.41 End-to-end capacity for nonparametric topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and long bit duration

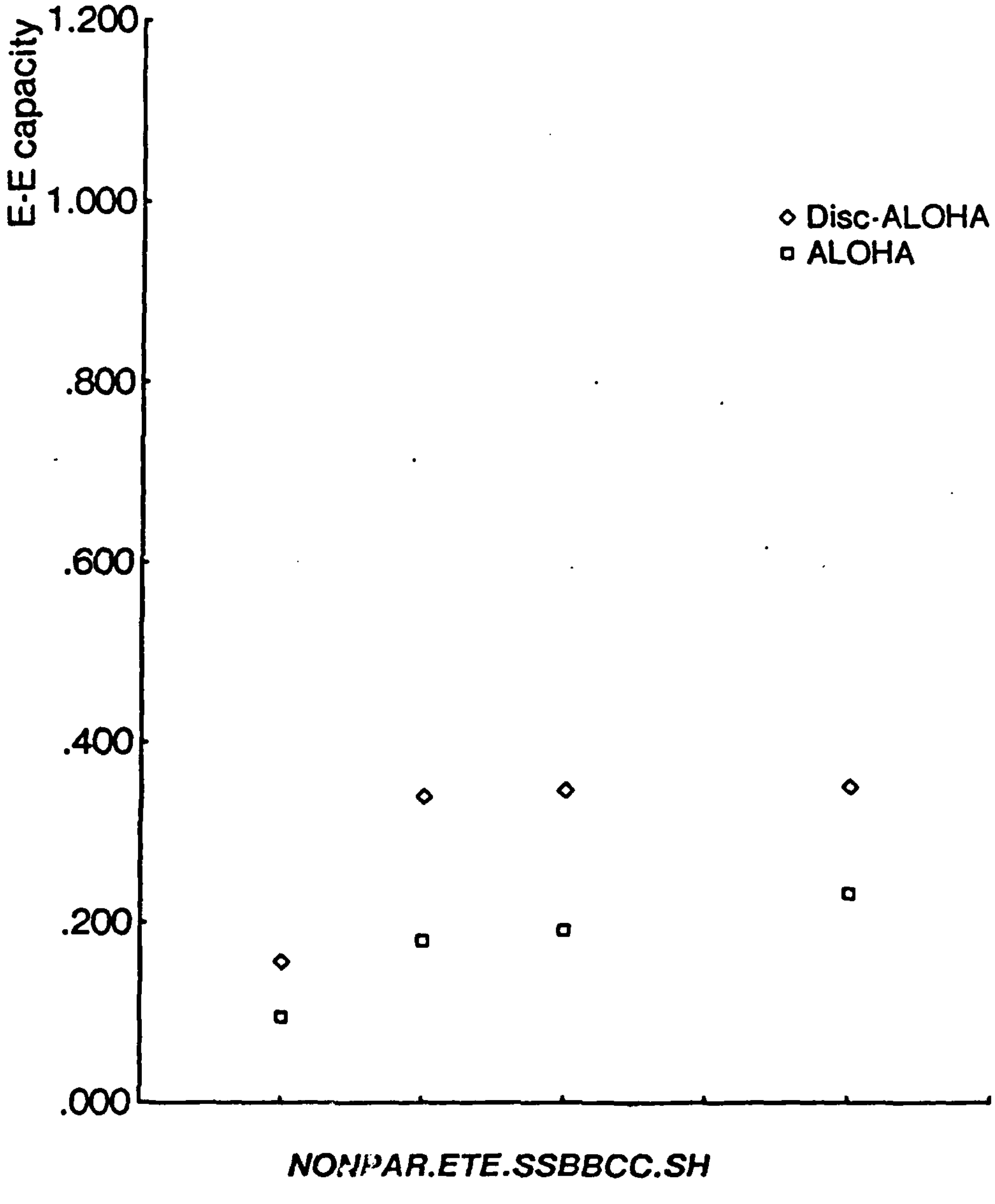


Fig. 9.42 End-to-end capacity for nonparametric topologies with uniform end-to-end traffic, for a spread spectrum system with bit-changing codes and short bit duration

Chapter 10

FINAL REMARKS

10.1 Conclusions

We presented in this work a model for the capacity analysis of multihop packet radio networks. This model is applicable to a large class of channel access protocols and modes of capture. The class of protocols that can be accommodated includes CSMA, a number of Busy Tone protocols, Disciplined-ALOHA, and ALOHA. The model includes as parameters the probability of bit error, the probability of synchronizing onto new packets, and the probability of collision caused by packet overlaps at a receiver. By the appropriate setting of these parameters it is possible to represent different capture modes, such as zero capture, and the type of capture typical of spread spectrum systems.

The assumptions made in the formulation of the model lead to Markovian stochastic processes. From the stationary measures of these processes and some appropriately defined passage times we derived throughput equations. We also pre-

sented algorithms and data structures for the evaluation of these expressions, and for the determination of the capacity associated with a given traffic pattern.

Different protocols have different requirements for the state information necessary for the description of their actions. We categorized the protocols in classes according to the complexity of the state description required, and for each such class formulated the models with the minimum necessary complexity. Some protocols lead to a product form solution for the stationary probability distribution of the associated stochastic model. The existence of a product form solution was shown to be equivalent to symmetry in the link blocking. This result was also shown to hold in the case where the packet lengths have an arbitrary distribution (with some mild restrictions). We also investigated the conditions under which a product form solution exists when the scheduling delay distribution has a general distribution. Even though the existence of a product form solution represents a simplification in the application of the analysis, we showed that nevertheless its computation is NP-hard.

The analytical model was applied to a number of topologies and system configurations in order to do a comparative evaluation of the performance of different channel access protocols. The systems considered were a narrowband system, a spread spectrum system with bit-changing codes, and spread spectrum systems with bit-homogeneous codes and uniform, receiver-directed, and transmitter-assigned code sequences. The results of the analysis showed that, over the situations examined, CSMA performed consistently better than Disciplined-ALOHA, which in turn performed consistently better than ALOHA. In the narrowband systems C-BTMA was seen to perform better than ID-BTMA, and the latter better than LD-BTMA. In these systems the performances of CSMA and Disciplined-ALOHA were comparable. The spread spectrum captured offered, over the zero capture of the narrowband

systems, a marked improvement in the performance of D-ALOHA and a not very significant improvement in the performance of ALOHA. For the spread spectrum systems with bit-homogeneous codes, where carrier sensing is feasible, CSMA performance showed a marked improvement over D-ALOHA in the situations where the number of hidden terminals is small. The relative performance of the Busy Tone protocols did not show the same fixed ordering as in the narrowband system, with ID-BTMA or LD-BTMA performing better than C-BTMA in a number of situations. In some cases CSMA also showed a better performance than C-BTMA. We examined systems with two different codeword lengths, and observed the reduced performance of the systems with the shorter codewords, due to the effects of (i) reduced signal-to-thermal-noise ration, and (ii) reduced protection against multiuser interference. Two distinct traffic patterns were considered for the cases analyzed. The relative protocol performance showed to be insensitive with respect to the type of traffic pattern.

10.2 Open Problems

Many more questions than answers exist concerning the performance of multihop packet radio networks. In connection with the model presented, one such question resides in finding product form approximations for the computation of throughput or, even better, approximations possessing polynomial time complexity. One such approximation was given by Chen and Boorstyn in [Chen85] for Disciplined ALOHA, but approximations for other protocols do not exist. Outside of the immediate scope of this model many other questions exist, some of which of a fundamental nature. One such question concerns the conjecture made in Section 1.3 that the boundary of the feasible region of the heavy-traffic throughputs, dealt

with in this work, represents the capacity of the system with infinite queues, in the sense that if S_0 is a point on that boundary then any set of link traffic requirements αS_0 is attainable with finite average queue sizes if $0 \leq \alpha < 1$, and is not attainable if $\alpha > 1$.

Another open problem concerns the queueing delay behavior of multihop packet radio systems. Except for very simple cases, no analytical results exist for systems of interfering queues, and the existing analytical tools do not allow one to go much further in this respect. An alternative to this situation might be the development of approximations. These approximations may require some additional insight into the problem, obtained from simulation studies. A possible approach for such approximations could be to assume some functional relationship between the average queue size at a node and the arrival rate at that queue, parameterized by the heavy-traffic capacity for the given *system parameters*.

Another problem for which results do not exist for the multihop case is the effect of nonzero propagation delay on the system performance. As discussed in Chapter 1, the exact analysis presents great difficulties, and the only alternative appears to be the development of approximations. The existence of a nonzero propagation delay makes it possible for a packet radio system to enter states that would not be allowed under zero propagation delay operation (for example, the state where two neighboring nodes are simultaneously transmitting, under CSMA). These transitions occur at the beginning of the packet transmission, during a "vulnerable period" with a duration equal to the propagation delay. A possible approximation would be to consider that these transitions can occur throughout the duration of the packet, but at a smaller rate. This rate would be adjusted so as to yield the same probability of transition to the "forbidden" state during a packet duration as the situation where such transitions only occur during the initial "vulnerable

period." In this way a Markovian model similar to the one described in this work can be constructed, although with an even larger state space, and thus more severe limitations from the point of view of the computational complexity.

Aside from the above mentioned modeling problems, many questions exist concerning the system behavior of packet radio networks. One such question is routing. Contrarily to the situation in point-to-point networks, in packet radio networks there is interaction between the routing and the capacity of the network links. Given a set of end-to-end traffic requirements, one would want to determine whether there exists some routing function for which the resulting link traffics are feasible and, if so, which such function minimizes the average packet delay. Transmit power control is also an important issue in packet radio. Any choice of the transmit power must perform a tradeoff between the number of hops needed for a packet to reach the final destination, and the interference that such transmissions cause at other receivers. A number of papers, in particular [Klei78], [Silv80], [deSo85], [Sous85], and [Hou86], have addressed this problem. Another important issue is the effect of forward error correction on the performance of packet radio networks. Storey and Tobagi present some results on this topic in [Stor85].

Efficient network operation requires adaptive real-time control procedures. An important issue in network control is the adaptive distributed control of the rescheduling parameters in order to achieve optimum delay performance satisfying the traffic constraints. To the best of our knowledge, no work has at the time of this writing been reported in the literature concerning this problem. For mobile operation another important question is that of network management, with aspects such as the maintenance and updating of connectivity and routing information. Some results are reported on this subject in [Garc85].

Appendix I

MULTIUSER INTERFERENCE

It is our goal in this Appendix to obtain a qualitative description of the multiuser interference in "typical" spread spectrum systems. Due to the analytical difficulty of the problem we shall not attempt to obtain absolute accuracy in the results derived. We would like, however, that the model that we construct represent well the relative changes, due to changes in system parameters such as received power levels and length of the code waveforms, in the quantities of interest associated with the multiuser interference, such as bit errors and average length of the successful packets. We would also like to avoid having to deal with the properties, in particular the correlation structure, of specific sets of code waveforms. In order to satisfy these requirements, we take the approach of imposing a probabilistic structure on the set of all code waveforms and studying the stochastic properties of the received signals derived therefrom. In this way we shall compute averages, over that set, of the individual results concerning each of the code waveforms in the set. (This is an idea similar to the one behind the "random coding" argument of Information Theory.) The averages thus obtained can be taken as lower bounds on the performance achievable with *ad hoc* selected code waveforms. In the model

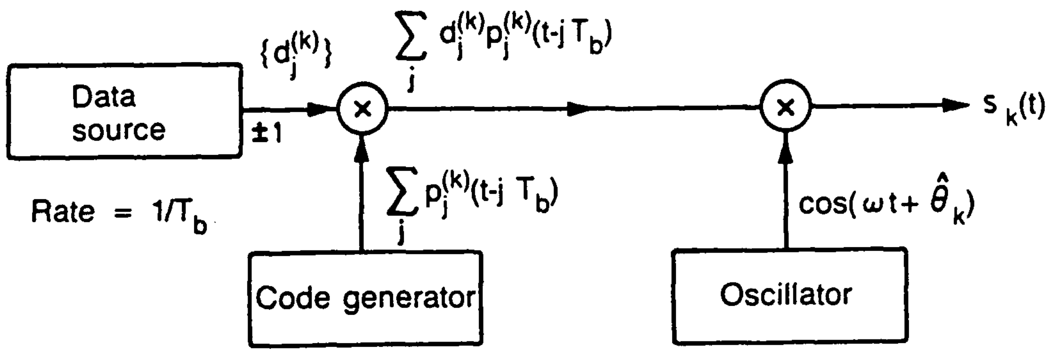
we formulate, we consider the chip amplitudes within each code waveform to form an independent and identically distributed (i.i.d.) sequence, of length N_c, of ± 1 random variables, where N_c is the given codeword length. By specifying the type of correlation between chip amplitudes of different codewords of a same transmission we shall be able to model both the cases of bit-homogeneous and bit-changing codes. Similarly, by specifying the type of correlation between the codewords assigned to different users we shall be able to model the uniform, receiver-directed, and transmitter-assigned types of code assignments.

Section I.1 describes the structure of the spread spectrum transmitter and receiver considered. Section I.2 studies the components of the signal at the output of the receiver due to the desired signal, the thermal noise, and single-user interference. Section I.3 studies the multiuser interference, and derives a capture model. This capture model includes expressions for the bit error probability and the probability of packet loss.

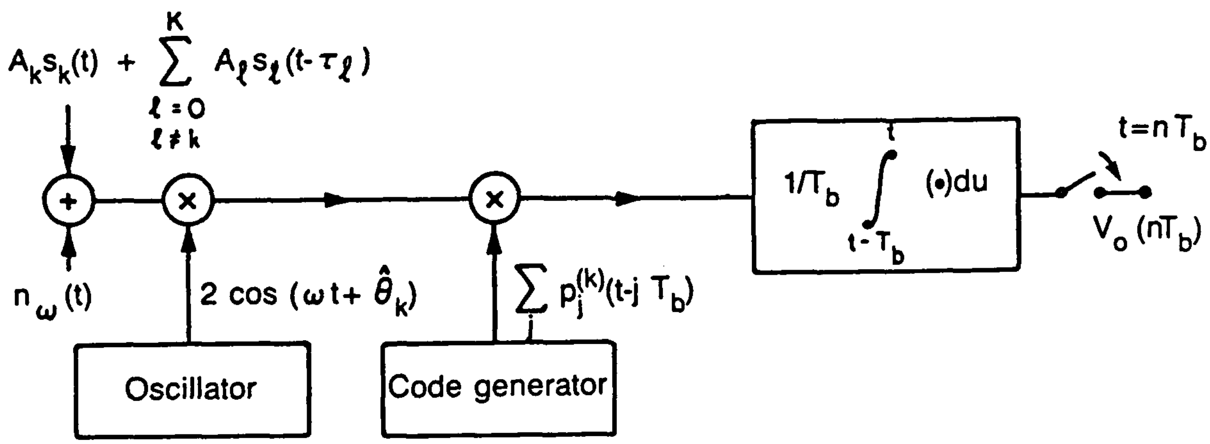
I.1 System Structure

We consider in this Section a direct sequence binary phase shift keying system, with $K+1$ transmitters and $K+1$ receivers, numbered $0, 1, \ldots, K$. The k-th transmitter sends a signal which is received by the k-th receiver, assumed to have the knowledge of the codes used in the transmission. The k-th receiver has also present at its input K other signals that interfere with the reception of the desired signal. This reception is in addition corrupted by thermal noise that we can assume added at the input of the receiver.

Transmitter structure and operation: We show in Figure I.1a the structure of the transmitter for the k-th signal. The transmitter comprises the source of data, the



(a)



(b)

Fig. I.1 Structure of direct-sequence transmitter and receiver

spread-spectrum modulator, and the BPSK modulator. The data source outputs a stream of binary digits, assumed to be a sequence of i.i.d. random variables taking the values +1 and −1 with equal probability, at a rate of $1/T_b$ digits/sec. We let $\{d_j^{(k)} : j = \ldots, -1, 0, 1, \ldots\}$ be the corresponding binary sequence. Each digit $d_j^{(k)}$ is multiplied, in the spread-spectrum modulator, by a code waveform $p_j^{(k)}(t)$. The code waveform $p_j^{(k)}(t)$ is formed by a sequence of N_c "chip" pulses, taking on the values +1 and −1 with equal probability. Within a codeword, the chip amplitudes are i.i.d. random variables taking on the values +1 and −1 with equal probability. Without loss of generality, the chip pulses are considered to have a rectangular shape. Letting $\Pi_\tau(t)$ denote a square τ-second pulse of unit amplitude defined by

$$\Pi_\tau(t) = \begin{cases} 1, & \text{if } 0 \le t \le \tau, \\ 0, & \text{otherwise,} \end{cases}$$

$T_c = T_b/N_c$ be the chip duration, and $a_{jN_c+l}^{(k)}$ be the amplitude of the l-th chip of codeword $p_j^{(k)}(t)$, we have

$$p_j^{(k)}(t) = \sum_{l=0}^{N-1} a_{jN_c+l}^{(k)} \Pi_{T_c}(t - lT_c).$$

The output of the spread spectrum modulator thus produces the spread-spectrum baseband signal $\sum_j d_j^{(k)} p_j^{(k)}(t - jT_b)$. This signal is then multiplied by a carrier waveform to produce a BPSK signal. We denote by $s_k(t)$ the unit-amplitude signal at the output of the k-th transmitter.

The choice of the correlation structure between different codewords of a given signal, or between the codewords of different signals, allows the modeling of the different types of code assignments, as follows.

1. *Bit-Homogeneous Codes:* The random code sequences assigned to different bits of a same transmission are identical. Formally, $p_j^{(k)}(t) \equiv p^{(k)}(t)$, for all j. In

terms of the individual chip amplitudes, this condition means that , for all j, $a_{jN_c+l}^{(k)} \equiv a_l^{(k)}$, where $\{a_l{}^{(k)}\}_{l=0}^{N_c-1}$ is the i.i.d. chip sequence defining the codeword. If signals $s_k(t)$ and $s_m(t)$ use different codewords, then $a_l^{(k)}$ and $a_n^{(m)}$ are independent, for $0 \leq l, n \leq N_c - 1$. If the signals use the same codeword, then $a_l^{(k)} = a_l^{(m)}, l = 0, \ldots, N_c - 1$.

2. *Bit Changing Codes:* The chip amplitudes assigned to different chips $a_l^{(k)}$ and $a_n^{(k)}$, $l \neq k$, of the same transmission are independent. Again, if signals $s_k(t)$ and $s_m(t)$ use different codes, we take $a_l^{(k)}$ and $a_n^{(m)}$ to be independent for all l and n, and otherwise we set $a_l^{(k)} = a_l^{(m)}$.

Receiver structure and operation: The receiver is shown in Figure I.1b. It comprises a local oscillator, a local code generator, and a data detector. BPSK demodulation is achieved by performing the product of the received signal with the local carrier, the double-frequency components being removed by the ensuing integrator. The despreading operation is performed by multiplying the resulting baseband spread spectrum signal by a locally generated replica of the desired signal code sequence. The product signal is then integrated over one bit duration. The output of the integrator is the crosscorrelation, computed over a bit duration, between the input signal and the local code waveform. This output is sampled at the bit boundaries, and the sampled value is compared with a threshold, in order to determine the value of the received data bit.

I.2 Receiver Output Signal

Without loss of generality, we focus on the reception of signal $s_0(t)$ by the corresponding receiver. The receiver has at its input attenuated and delayed versions $A_k s_k(t - \tau_k)$ of the transmitted signals $s_k(t)$, $k = 0, 1, \ldots, K$, as well as a zero mean

white Gaussian noise of two-sided power spectral density $\eta_0/2$. We take the time delay of $s_0(t)$ to be $\tau_0 = 0$, and assume the bit boundaries of the interfering signals to be uniformaly distributed within the bit duration of the signal $s_0(t)$. We also assume that each receiver is in perfect code, carrier frequency, and carrier phase synchronism with the signal it is supposed to receive. We let θ_k designate the phase of the carrier of signal $s_k(t)$ at the input of the zeroth receiver. We take $\theta_0 = 0$, and assume $\{\theta_i\}_{i=1}^{K}$ to be a set of i.i.d. random variables uniformly distributed in $(-\pi, \pi)$.

The receiver output is a linear functional of the input, and thus we can study separately the components of the output signal corresponding to each one of the input signals. Without loss of generality, we consider the output of the receiver at time $t = T_b$, corresponding to the reception of bit $d_0^{(0)}$ of signal $s_0(t)$.

I.2.1 Desired Signal

The output at $t = T_b$ due to the desired signal is

$$v_0(T_b) = \frac{1}{T_b}\int_0^{T_b} A_0\, d_0^{(0)} 2\cos^2(\omega t)\left(\sum_j p_j^{(0)}(t - jT_b)\right)^2 dt\,.$$

Since the amplitude of the code waveforms is of unit absolute value, the above expression becomes

$$\begin{aligned} v_0(T_b) &= \frac{1}{T_b}\int_0^{T_b} 2\, A_0\, d_0^{(0)} \cos^2(\omega t)\, dt \\ &= A_0\, d_0^{(0)}\left[1 + \frac{\sin(2\omega T_b)}{2\omega T_b}\right] \\ &\simeq A_0\, d_0^{(0)}, \end{aligned}$$

having in the last expression neglected the terms of frequency 2ω, since normally $\omega T_b << 1$. The output due to the desired signal is seen to be, as expected, an attenuated version of the transmitted data bit $d_0^{(0)}$.

I.2.2 Thermal Noise

The output signal due to the thermal noise is

$$v_0(T_b) = \frac{1}{T_b} \int_0^{T_b} 2\, n_w(t)\, p_0^{(0)}(t) \cos(\omega t) dt\,.$$

Since $p_0^{(0)}(t)$ only takes the values ± 1, $n(t) = p_0^{(0)}(t)\, n_w(t)$ is also a white Gaussian noise with the same power spectral density $\eta_0/2$. Thus $v_0(T_b)$ is a Gaussian random variable, with

$$E[v_0(T_b)] = 0$$

and variance

$$\begin{aligned}
\mathrm{var}[v_0(T_b)] = E[v_0^2(T_b)] &= \frac{4}{T_b^2} E\left[\int_0^{T_b}\int_0^{T_b} n(t)\, n(u) \cos(\omega t) \cos(\omega u)\, dt\, du\right] \\
&= \frac{\eta_0}{T_b}\left[1 + \frac{sin(2\omega T_b)}{2\omega T_b}\right] \\
&\simeq \frac{\eta_0}{T_b},
\end{aligned}$$

having again neglected the double-frequency components.

I.2.3 Single Interfering Signal

We consider now the output signal due to a generic interfering signal $r(t)$. For notational simplicity, we shall remove all superscripts from the parameters associated with this signal, so that $r(t) = A\cos(\omega t + \theta) \sum_j d_j p_j(t - jT - \tau)$, where $p_j(t)$

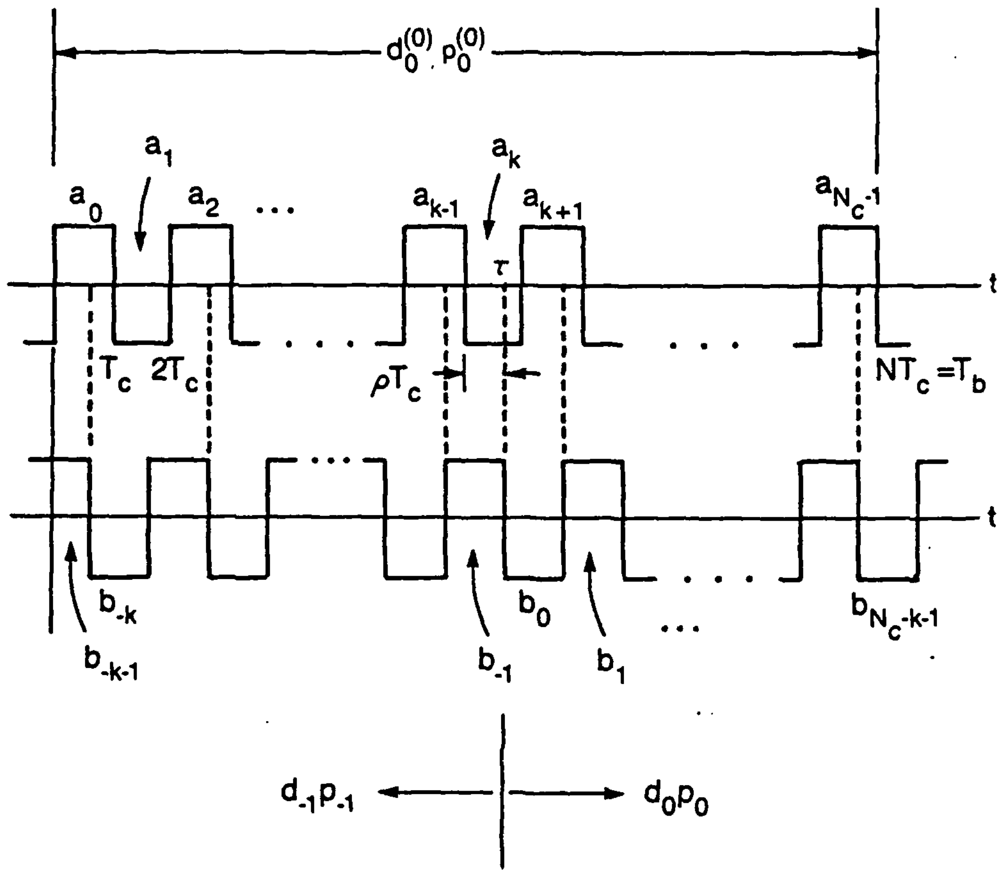


Fig. I.2 Chip waveforms of interfering and local codes

is the j-th codeword and d_j the j-th data bit of the interfering signal. We write $\tau = kT_c + \rho T_c$, where k is an integer determined by the condition that $0 \leq \rho < 1$. For simplicity, we shall consider only with the case $0 \leq \tau < T_b$ (i.e., $0 \leq k \leq N_c - 1$), since the other cases are easily derived from this one. Thus bit $d_0^{(0)}$ of the desired signal is overlapped by bits d_{-1} and d_0 of the interfering signal, with the boundary between d_{-1} and d_0 occurring during the k-th chip of codeword $p_0^{(0)}(t)$ (Figure I.2).

From Figure I.2 it is easy to see that the output voltage due to the interfering

signal is given by

$$v_0(T_b) = \frac{1}{T_b}\int_0^{T_b} A \sum_{j=-1}^{0} d_j\, p_j(t - jT_b - \tau)\, p_0^{(0)}(t)\, [\cos\theta + \cos(2\,\omega t + \theta)]\; dt$$

$$\simeq \frac{A}{T_b}\cos\theta \int_0^{T_b} [d_{-1}\, p_{-1}(t + T_b - \tau) + d_0\, p_0(t-\tau)]\; p_0^{(0)}\, dt\,,$$

having again neglected components of frequency 2ω. For notational simplicity, we designate by $\{a_k\}_{k=0}^{N_c-1}$, $\{b_k\}_{k=-N_c}^{-1}$, and $\{b_k\}_{k=0}^{N_c-1}$ the chip sequences forming the codewords $p_0^{(0)}(t)$ of the desired signal, and $p_{-1}(t)$ and $p_0(t)$ of the interfering signal, respectively. Given the assumption of rectangular chip pulses, the integral defining $v_0(T_b)$ becomes

$$\begin{aligned} v_0(T_b) = \frac{T_c}{T_b} A\cos\theta \Bigg\{ \rho \left[d_{-1} \sum_{l=0}^{k} a_l\, b_{-k-1+l} + d_0 \sum_{l=k+1}^{N_c-1} a_l\, b_{l-k-1} \right] \\ + (1-\rho) \left[d_{-1} \sum_{l=0}^{k-1} a_l\, b_{-k+l} + d_0 \sum_{l=k}^{N_c-1} a_l\, b_{l-k} \right] \Bigg\}, \end{aligned} \tag{I.1}$$

where we make the convention the a summation is to be taken as empty whenever the lower limit exceeds the upper limit. Note that the value of $v_0(t)$ for τ outside the interval $[0, T_b)$ can be obtained from the Equation (I.1) by substituting interfering data bits d_{-1} and d_0 by the interfering data bits which overlap bit 0 of the desired signal.

In the following analysis, we will make use of the following

Lemma I.3.1 *Let X, Y, and Z be i.i.d. random variables, taking on the values $+1$ and -1 with probability 1/2, and C_1 and C_2 be given constants taking either the value $+1$ or -1. Then $V_1 = C_1 X\, Y$ and $V_2 = C_2 X\, Z$ are independent random variables, taking on the values $+1$ or -1 with probability $1/2$.*

The form of $v_0(T_b)$ depends on the type of dependence between the codewords $p_0(t)$ and $p_1(t)$ of the interfering signal, and codeword $p_0^{(0)}$ of the desired signal. We first consider the case where the desired and the interfering signals use different code sequences, modeled by taking the codes of the interfering signal to be independent from those of the desired signal. We then consider the case where the code sequences of the two signals are identical, for both the subcases of bit-changing and bit-homogeneous code sequences.

I.2.3.1 Different Codes

According to the model of Section I.2, the case of different code sequences is handled by taking a_k and b_l to be independent random variables, for all values of k and l. Accordingly, and from Lemma I.3.1, any two distinct terms inside the braces in Equation (I.1) are independent, so that we can write

$$v_0(T_b) = \frac{T_c}{T_b} A\cos\theta \left\{ \rho \sum_{l=0}^{N_c-1} X_l + (1-\rho) \sum_{l=0}^{N_c-1} Y_l \right\},$$

where $\{X_l\}$ and $\{Y_l\}$ are sets of i.i.d. random variables taking on the values $+1$ and -1 with probability $1/2$. By writing

$$v_0(T_b) = \frac{A\cos\theta}{\sqrt{N_c}} \left\{ \rho \frac{\sum_{l=0}^{N_c-1} X_l}{\sqrt{N_c}} + (1-\rho) \frac{\sum_{l=0}^{N_c-1} Y_l}{\sqrt{N_c}} \right\}$$

and using the Central Limit Theorem we see that, for large N_c, $v_0(T_b)$ is the weighted sum of two independent approximately normal random variables, and is thus itself approximately normal. Given θ and ρ, we can thus approximate $v_0(T_b)$ by a normal random variable with mean $\mu = 0$ and variance

$$\sigma^2 = \frac{A^2\cos^2\theta}{N_c} \left[\rho^2 + (1-\rho)^2\right].$$

However, and unlike a normal random variable, $v_0(T_b)$ is bounded ($|v_0(T_b)| \leq A$) with probability 1. The nonzero probability mass associated with the tails of the normal random variable outside the interval $[-A, A]$ will cause the approximation to be pessimistic, but becoming more accurate as σ/A decreases, that is, as the codeword length N_c increases.

It should be noted that, unconditioned on θ and ρ, $v_0(T_b)$ is not normal, and its distribution is difficult to compute. It is easy, however, to compute the two first moments,

$$E[v_0(T_b)] = 0\,,$$

and

$$E\left[v_0^2(T_b)\right] = E\left[E\left[v_0^2(T_b) \mid \rho, \theta\right]\right] = E\left[\frac{A^2 \cos^2\theta}{N_c}\left[\rho^2 + (1-\rho)^2\right]\right] = \frac{A^2}{3\,N_c}.$$

The dependence between the values of the interference output voltage for different bits of a same transmission depends on the type of the code sequence. If bit-homogeneous codes are used and no thermal noise is present, $v_0(n\,T_b)$ can only take four distinct values over a given sample path, corresponding to the four distinct combinations of values of d_{-1} and d_0. If bit-changing codes are used, then $v_0(T_b), v_0(2\,T), \ldots$, are independent.

I.2.3.2 Identical Bit-Homogeneous Codes

In this case, $a_k = b_{-jN_c+k}$, for all j integer, and $k = 0, 1, \ldots, N-1$. Equation (I.1) then becomes

$$v_0(T_b) = \frac{T_c}{T_b} A \cos\theta \left\{ \rho \left[d_{-1} \sum_{l=0}^{k} a_l\, a_{N_c-k-1+l} + d_0 \sum_{l=k+1}^{N_c-1} a_l\, a_{l-k-1} \right] \right.$$
$$\left. + (1-\rho) \left[d_{-1} \sum_{l=0}^{k-1} a_l\, a_{N_c-k+l} + d_0 \sum_{l=k}^{N_c-1} a_l\, a_{l-k} \right] \right\}.$$

The form of the interfering voltage depends now on the value of k.

(i) $k = 0$

We have

$$v_0(T_b) = \frac{T_c}{T_b} A\cos\theta \left\{ \rho \left[d_{-1}\, a_0\, a_{N_c-1} + d_0 \sum_{l=1}^{N_c-1} a_l\, a_{l-1} \right] + (1-\rho)\, d_0\, N_c \right\}$$

$$= A\cos\theta\,(1-\rho)\, d_0 + \rho\, \frac{A\cos\theta}{\sqrt{N_c}}\, \frac{\sum_{l=0}^{N_c-1} X_l}{\sqrt{N_c}},$$

where $\{X_l\}$ is an i.i.d. sequence of binary equiprobable random variables. We thus see that $v_0(T_b)$ has a deterministic component

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = A\cos\theta\, d_0(1-\rho)$$

and a zero mean approximately normal component of variance

$$\mathrm{var}\left[v_0(T_b) \mid k, \rho, \theta\right] = \frac{A^2\cos^2\theta}{N_c}\, \rho^2.$$

(ii) $k = N_c - 1$

Performing similar computations, one would arrive at the conclusion that $v_0(T_b)$ has a deterministic component

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = A\cos\theta\, d_{-1}\, \rho$$

and a zero mean approximately normal component of variance

$$E\left[v_0^2(T_b) \mid k, \rho, \theta\right] = \frac{A^2\cos^2\theta}{N_c}\,(1-\rho)^2.$$

(iii) $1 \le k \le N_c - 2$

In this case we have

$$v_0(T_b) = \frac{A\cos\theta}{N_c} \left\{ \rho \sum_{l=0}^{N_c-1} X_l + (1-\rho) \sum_{l=0}^{N_c-1} Y_l \right\}$$

and thus $v_0(T_b)$ is approximately normal, with mean

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = 0$$

and variance

$$E\left[v_0^2(T_b) \mid k, \rho, \theta\right] = \frac{A^2 \cos^2\theta}{N_c}\left[\rho^2 + (1-\rho^2)\right].$$

Cases (i)–(iii) deal only with the case where $0 \leq \tau < T_b$. Due to the periodicity of the codes involved, the results for τ outside the range can be immediately obtained from those derived above by shifting appropriately the indices of the interfering data bits that overlap bit $d_0^{(0)}$ of the desired signal. We show in Figures I.3a and I.3b, respectively, the deterministic component and the variance of the random component of $v_0(T_b)$ as function of τ. In Figure I.3a we also show the bit amplitudes that affect each of the code waveforms correlation peaks. These correlation peaks correspond to the "vulnerable" periods mentioned in Section 2.3.4, such that if the timing of the interfering signal is within them, a high value of the interfering voltage results.

I.2.3.3 Identical Bit-Changing Codes

This situation is modeled by taking $a_k = b_k$ for all k, with a_k, a_j independent for $k \neq j$. As in the previous case, the behavior of $v_0(T_b)$ depends on k. Since the derivations are similar, we limit ourselves to state the results.

(i) $k = 0$

The output $v_0(T_b)$ has a deterministic component

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = A \cos\theta \, d_0 \,(1-\rho)$$

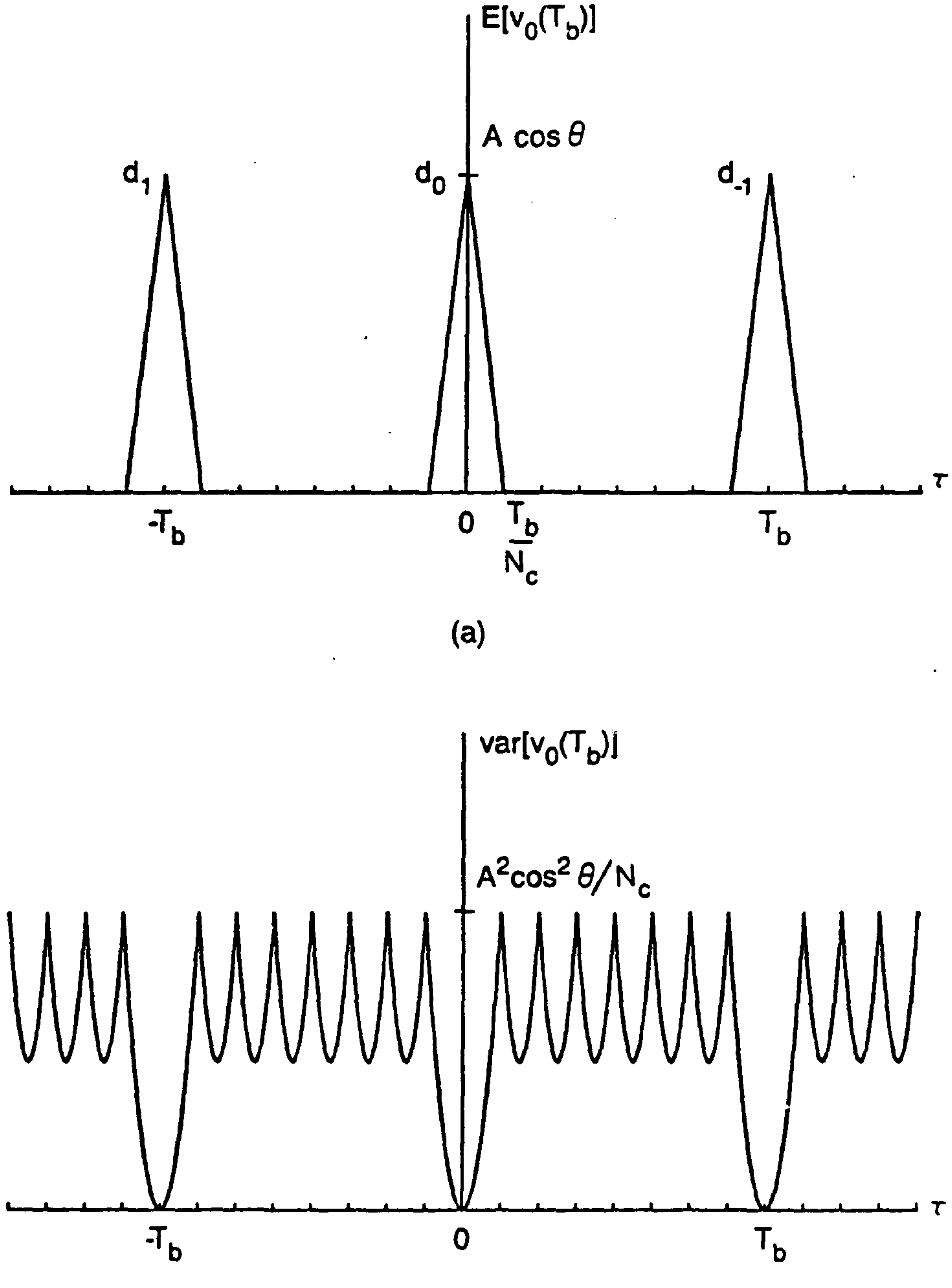


Fig. I.3 Mean and variance of interference for identical bit-homogeneous codes

and a zero mean random component, with variance

$$E\left[v_0^2(T_b) \mid k, \rho, \theta\right] = \frac{A^2 \cos^2 \theta}{N_c} \rho^2.$$

Again, for large N_c, we can take the random component to be approximately normal.

(ii) $k = -1$

We have a situation similar to case (i), but now with

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = A \cos\theta \, d_0 \, \rho$$

and

$$E\left[v_0^2(T_b) \mid k, \rho, \theta\right] = \frac{A^2 \cos^2 \theta}{N_c} (1 - \rho)^2.$$

(iii) $k \neq 0$, $k \neq -1$

For $k \neq 0$ or $k \neq -1$, each interfering chip is independent of the chips of the desired codeword that it overlaps, and we are thus in a situation similar to that of independent codes. We have

$$E\left[v_0(T_b) \mid k, \rho, \theta\right] = 0$$

and

$$E\left[v_0^2(T_b) \mid k, \rho, \theta\right] = \frac{A^2 \cos^2 \theta}{N_c} \left[\rho^2 + (1 - \rho)^2\right].$$

We show in Figures I.4a and I.4b, respectively, the deterministic component and the variance of the random component of $v_0(T_b)$ as function of τ. As observed in Section 2.3.4, the vulnerable period during which the interfering and desired codes possess high correlation is limited to one chip time around $\tau = 0$.

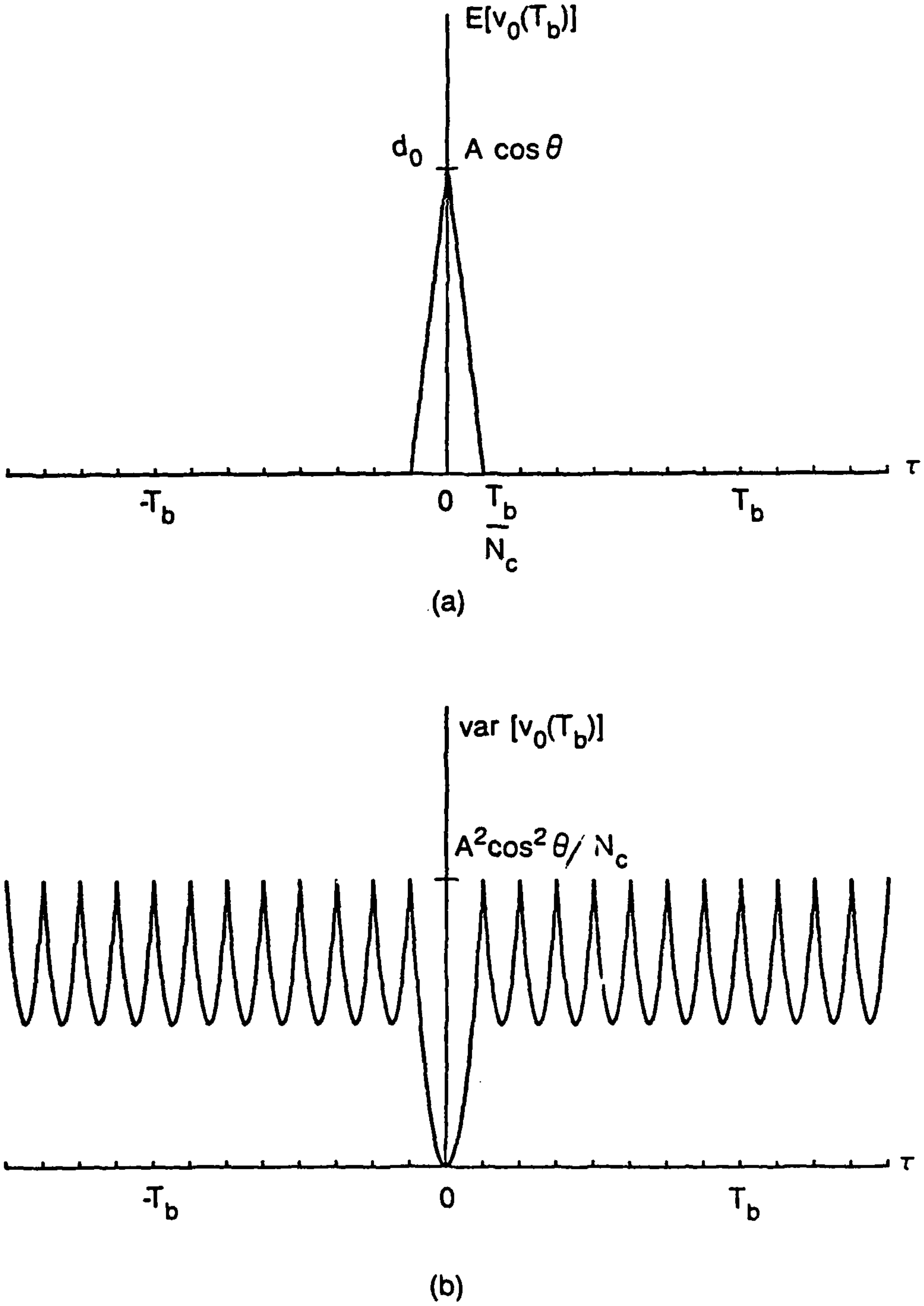


Fig. I.4 Mean and variance of interference for identical bit-changing codes

I.3 Multiuser Interference: Bit Errors and Packet Loss

We now consider a general situation where a desired signal $s_0(t)$ is overlapped by p signals using the same code sequence, denoted $s_1(t), \ldots, s_p(t)$, and by q signals using independent code sequences, denoted $s_{p+1}, \ldots, s_{p+q}(t)$. In addition, a zero mean white gaussian noise of (two-sided) spectral density $\eta_0/2$ is present. The receiver is assumed to be in frequency, phase and code synchronism with the desired signal. Signal $s_k(t)$, $k = 1, \cdots, p+q$, has differential carrier phase θ_k, and differential time delay τ_k, relative to the corresponding quantities of signal $s_0(t)$.

The amplitude $A_k, k = 0, \cdots, p+q$, of the signals present, as well as η_0, are assumed to be known a priori, and $\{\theta_k\}_{k=1}^{p+q}, \{\tau_k\}_{k=1}^{p+q}$ are assumed to be independent random variables, uniformly distributed in $(-\pi, \pi)$ and $(0, T_b)$, respectively. We will consider in separate both the cases of bit-changing codes and bit-homogeneous codes.

Let $\mathbf{A}$ be the vector of the signal amplitudes, and $\mathbf{T}$ and $\mathbf{\Theta}$ be the vectors of differential time delays and differential carrier phases, respectively, with respect to signal $s_0(t)$. Let also $I(\mathbf{A}, \mathbf{T}, \mathbf{\Theta})$ be the interference component of the receiver output at time $t = T_b$, not including the thermal noise. We have

$$I(\mathbf{A}, \mathbf{T}, \mathbf{\Theta}) = \sum_{k=1}^{p+q} I_k(A_k, T_k, \theta_k)$$

where $I_k(A_k, T_k, \theta_k)$ is the interference due to signal $s_k(t)$, and has the general form given by Equation (I.1). For reasons of symmetry, the bit error probability does not depend on which data bit is transmitted. Assume that desired data bit -1 is transmitted, producing a component $-A_0$ at the receiver output. An error in the reception of the data due to the thermal noise and multiuser interference will occur

with probability

$$P_{ec}(\mathbf{A},\mathbf{T},\Theta) = P(e \mid \mathbf{A},\mathbf{T},\Theta) = P(-A_0 + n_w + I(\mathbf{A},\mathbf{T},\Theta) > 0)$$
$$= P(n_w + I(\mathbf{A},\mathbf{T},\Theta) > A_0), \tag{I.2}$$

where n_w is the output component due to the thermal noise. This probability depends on the statistics of the multiuser interference, which we now examine.

I.3.1 Bit-Changing Codes

For bit-changing codes, the difference between the type of interference due to signals with the same codes as the desired signal and signals with different (independent) codes resides in the existence, in the former case, of one single "vulnerable period" of width double that of the chip duration T_c such that if the timing of an interfering signal falls within this period, a large interference output results. For typical values of codeword length of 100 chips/bit, and packet length of 1000 bits, this duration represents a fraction of 2×10^{-5} of the packet length, and will be neglected. Thus we assume that both types of signals produce the same type of interference. For the large typical values of N_c, we assume each component of the interference $I(\mathbf{A},\mathbf{T},\Theta)$ to be Gaussian, with zero mean and variance

$$\sigma_k^2 = \frac{A_k^2 \cos^2 \Theta_k}{N_c} [\rho_k^2 + (1-\rho_k)^2].$$

With this assumption, $I(\mathbf{A},\mathbf{T},\Theta)$ is itself Gaussian, with zero mean and variance

$$\sigma^2 = \frac{1}{N_c} \sum_{k=1}^{p+q} A_k^2 \cos^2 \Theta_k [\rho_k^2 + (1-\rho_k)^2].$$

To the code interference is also added the thermal noise interference n_w which, from Section I.3.2, is Gaussian with zero mean and variance η_0/T_b. Thus the total interference voltage at the receiver output is Gaussian, with zero mean and variance

$$\sigma_i^2 = \frac{\eta_0}{T_b} + \frac{1}{N_c}\sum_{k=1}^{p+q} A_k^2 \cos^2\Theta_k \left[\rho_k^2 + (1-\rho_k)^2\right].$$

The approximation of assuming a Gaussian distribution becomes more accurate as η_0/T_b increases over σ^2, which happens either as the noise power increases or as the codeword length increases. From Equation I.2 we obtain for the probability of error, conditioned on $\mathbf{T}$ and Θ,

$$P_{ec}(\mathbf{A},\mathbf{T},\Theta) = Q\left(\sqrt{\frac{A_0^2}{\frac{\eta_0}{T_b} + \frac{1}{N_c}\sum_{k=1}^{p+q} A_k^2 \cos^2\Theta_k \left[\rho_k^2 + (1-\rho_k)^2\right]}}\right),$$

where $Q(x)$ is defined by

$$Q(x) \triangleq \int_x^\infty \frac{e^{-y^2/2}}{\sqrt{2\pi}}\,dy\,.$$

The probability of bit error is then given by

$$P_e(\mathbf{A}) = E[P_{ec}(\mathbf{A},\mathbf{T},\Theta)].$$

The computation of expectations of this type constitutes one of the central points in the study of multiuser interference, about which a substantial body of work exists in the spread spectrum literature (e.g., [Yao77], [Purs77a], [Purs77b], [Purs82], [Gera82]). Even with the simple forms of the distributions assumed for $\{\Theta_k\}$ and $\{\tau_k\}$ this computation is difficult, and we shall not attempt it here. We will instead consider a worst-case bound $P_e^*(\mathbf{A})$ for $P_e(\mathbf{A})$, obtained by assuming perfect chip

and carrier synchronism (i.e., $\tau_k = 0$ and $\theta_k = 0, k = 1, \ldots, p+q$) at the receiver between the interfering signals and the desired signal. Thus

$$P_e^*(\mathbf{A}) = Q\left(\sqrt{\frac{A_0^2}{\frac{\eta_0}{T_b} + \frac{1}{N_c}\sum_{k=1}^{p+q} A_k^2}}\right). \tag{I.3}$$

Letting $(S/N)_0$ designate the basic signal-to-noise ratio in the absence of multiuser interference, $(S/N)_0 = A_0^2 T_b/\eta_0$, and letting P_k be the received power of signal $s_k(t)$, we can equivalently write

$$P_e^*(\mathbf{A}) = Q\left(\sqrt{\frac{(\frac{S}{N})_0}{1 + \frac{1}{N_c}(\frac{S}{N})_0 \sum_{k=1}^{p+q}(\frac{P_k}{P_0})}}\right). \tag{I.4}$$

The consideration of random code waveforms and the approximation afforded by the Central Limit Theorem led us to consider the multiuser interference to have a Gaussian distribution. This approximation is ofter made when dealing with deterministic code waveforms, being referred to as the *Gaussian interference* approximation. For some specific sets of code waveforms it has been shown to yield good results when the number of interfering signals is large, the code lengths are large, or the thermal noise power is high ([Yao77]). These conditions coincide with those given here for the distribution of the interference signal $I(\mathbf{A}, \mathbf{T}, \Theta)$ to be approximately normal.

In the model considered, in which coding is not present, a packet is successful only if no bit errors occur. In the present case of bit-changing codes bit errors are independent from bit to bit, and thus the probability of successful reception for a

packet of length N is $(1 - P_e^*(\mathbf{A}))^N$. In the continuous-time model considered in this work, errors are assumed to form a Poisson point process with rate $P_e^*(\mathbf{A})/T_b$.

I.3.2 Bit-Homogeneous Codes

As in the previous case, the behavior of the multiuser interference depends on the relationship between the code sequences of the interfering and desired signals. The computation of the conditional probability of bit error involves the computation of the conditional probability of bit error $P_{ec}(\mathbf{A}, \mathbf{T}, \mathbf{\Theta})$ and of the expectation $P_e(\mathbf{A}) = E[P_{ec}(\mathbf{A}, \mathbf{T}, \mathbf{\Theta})]$. We again take the worst-case situation where the interfering signals are in chip and carrier synchronism with the desired signal.

The interference due to the signals with different code sequences is taken to be Gaussian, with zero mean and variance $\frac{1}{N_c}\sum_{k=p+1}^{p+q} A_k^2$. The interference due to the signals $s_1(t), \cdots, s_p(t)$ with the same code sequence depends on the vector $\mathbf{T} = \{\tau_k\}$ of the differential time delays. If some T_k falls within any of the "vulnerable" intervals $[jT_b - T_c, jT_b + T_c]$, for j integer, an output component $A_k d_j^{(k)}$ will result, where $d_j^{(k)}$ takes the value $+1$ or -1 with probability $1/2$. For values of A_k comparable to A_0, which will be the case of the situation to be considered later for numerical study, we assume that any such component will give rise to a bit error with probability $1/2$. This is clearly a pessimistic assumption, since for more than one such interfering signal, the probability of the sum of n_w and the terms of the form $A_k d_j^{(k)}$ exceeding A_0 is less than $1/2$. The assumption is also not clearly valid in situations where some of the interfering signals have amplitudes much smaller than the desired signal, in which case a more refined model should be used. If no T_k falls within a "vulnerable period", the interference by signals $s_1(t), \cdots, s_p(t)$ is taken as Gaussian, as described in Section I.3.2. In this case, the total interference

is then Gaussian, with zero mean and variance

$$\sigma_t^2 = \frac{\eta_0}{T} + \frac{1}{N_c}\sum_{k=1}^{p+q} A_k^2,$$

to which corresponds a probability of error given by Equations (I.3) and (I.4).

In order to compute the probability of no bit errors, it is necessary to know the correlation between errors in different bits. If some τ_k, $k = 1, \cdots, p$, falls within a vulnerable period, the resulting bit errors are highly correlated, and the loss of the packet is assumed to result with probability 1. Otherwise, bit errors are assumed to occur independently from bit to bit. This assumption is just an approximation, since the multiuser interference is correlated from bit to bit. However, since thermal noise is added to it and the multiuser interference term becomes less important as N_c increases, this approximation will be accurate for large N_c.

We shall thus assume the following model

(i) with probability $1-(1-\frac{2}{N_c})^p$ a packet is lost due to interference within the vulnerable period;

(ii) with probability $(1-\frac{2}{N_c})^p$ no interference exists within the vulnerable period, but bit errors occur independently from bit to bit with probability $P_e^*(\mathbf{A})$; the bit errors are approximated by a Poisson point process with rate $P_e^*(\mathbf{A})/T_b$.

REFERENCES

[Abra70] N. Abramson, "The ALOHA System – Another Alternative for Computer Communications," *AFIPS Conf. Proc.*, vol. 37, FJCC, pp. 281–285, 1970.

[Abra77] N. Abramson, "The Throughput of Packet Broadcast Channels," *IEEE Trans. Comm.*, vol. COM-25, no. 1, pp. 117-128, Jan. 1977.

[Aho83] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983.

[Akav79] G. Akavia, L. Kleinrock, "Performance Tradeoffs and Hierarchical Designs of Distributed Packet-Switching Communication Networks," Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, *UCLA Tech. Report* UCLA-ENG–7952, Sept. 1979.

[Barb81] A. D. Barbour, R. Schassberger, "Insensitive Average Residence Times in Generalized Semi-Markov Processes," *Adv. Appl. Prob.*, vol. 13, pp. 720–735, 1981.

[Behr82] D. Behrman, W. C. Fifer, "A Low-Cost Spread-Spectrum Packet Radio," Proc. MILCOM'82, vol. 3, Boston, 1982.

[Boor80] R. R. Boorstyn, A. Kershenbaum, "Throughput Analysis of Multihop Packet Radio Networks," *Proc. ICC'80,* Seattle, Washington, Jun. 1980.

[Boor82] R. R. Boorstyn, A. Kershenbaum, V. Sahin, "A New Acknowledgment Protocol for Analysis of Multihop Packet Radio Networks," *Proc. COMPCON'82,* Washington, D.C., Sept. 1982.

[Braz84] J. M. Brázio, F. A. Tobagi, "Theoretical Results on the Throughput Analysis of Multihop Packet Radio Networks," *Proc. ICC'84,* Amsterdam, The Netherlands, May 1984.

[Braz85a] J. M. Brázio, F. A. Tobagi, "Throughput Analysis of Spread-Spectrum Multihop Packet Radio Networks," *Proc. INFOCOM'85,* Washington, D.C., Mar. 1985.

[Braz85b] J. M. Brázio, F. A. Tobagi, "Conditions for Product Form Solutions in Multihop Packet Radio Network Models," *SEL Technical Report* no. 85–274, Computer Systems Laboratory, Stanford University, Stanford, California, Apr. 1985.

[Chan77] K. M. Chandi, J. H. Howard, D. F. Towsley, "Product Form and Local Balance in Queueing Networks," *JACM,* vol. 24, no. 2, pp. 250–263, Apr. 1977.

[Chen85] M.-S. Chen, R. Boorstyn, "Throughput Analysis of CDMA Multihop Packet Radio Networks in the Presence of Noise," *Proc. INFOCOM'85,* Washington, D.C., Apr. 1985.

[Çinl75] E. Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[Cont72] S. D. Conte, C. de Boor, *Elementary Numerical Analysis*, McGraw-Hill, New York, 1972.

[Davi80] D. H. Davis, S. A. Gronemeyer, "Performance of Slctted ALOHA Random Access with Delay Capture and Randomized Time of Arrival," *IEEE Trans. Comm.*, vol. COM-28, no. 5, pp. 703–710, May 1980.

[deSo85] O. deSouza, M.-S. Chen, R. R. Boorstyn, "A Comparison of the Performance of Protocols in Packet Radio Networks," Proc. MILCOM'85, Boston, MA, Oct. 1985.

[Dixo84] R. C. Dixon, *Spread Spectrum Systems*, John Wiley, New York, 1984.

[Fayo79] G. Fayolle, R. Iasnogorodsky, "Two Coupled Processors – The Reduction to a Riemman-Hilbert Problem," *Z. Wahrscheinlichkeitstheorie*, vol. 47, pp. 325–351, 1979.

[Ferg77] M. J. Ferguson, "An Approximate Analysis of Delay for Fixed and Variable Length Packets in an Unslotted ALOHA Channel," *IEEE Trans. Comm.*, vol. COM-25, no. 7, Jul. 1977.

[Fral75] S. C. Fralick, J. C. Garrett, "Technological considerations for packet radio networks," *Proc. AFIPS*, pp. 233–243, 1975.

[Garc85] J. J. Garcia-Luna Aceves, N. Shacham, "Analysis of Routing Strategies for Packet Radio Networks," *Proc. INFOCOM'85*, Washington, DC, March 1985.

[Gare79] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.

[Gera82] E. A. Geraniotis,M. B. Pursley, "Error Probability for Direct-Sequence Spread-Spectrum Multiple-Access Communication—Part II: Approximations," *IEEE Trans. Comm.*, vol. COM-30, no. 5, pp. 985–995, May 1982.

[Gitm75] I. Gitman, "On the Capacity of Slotted ALOHA Networks and Some Design Problems," *IEEE Trans. Comm.*, vol. COM-23, no. 3, pp. 305-317, Mar. 1975.

[Hend83] W. Henderson, "Insensitivity and Reversed Markov Processes," *Adv. Appl. Prob.*, vol. 15, pp. 752–768, 1983.

[Heym82] D. Heyman, M. Sobel, *Stochastic Models in Operations Research, Volume 1*, McGraw Hill, New York, 1982.

[Hou86] T.-C. Hou, V. O. K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Comm.*, vol. COM-34, no. 1, pp. 38-44, Jan. 1986.

[Kahn78] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, R. C. Kunzelman, "Advances in Packet Radio Technology," *Proc. IEEE*, vol. 66, no. 11, pp. 1468–1496, 1978.

[Kell76] F. P. Kelly, "Networks of Queues," *Adv. Appl. Prob.*, vol. 8, pp. 416–432, 1976.

[Kell79] F. Kelly, *Reversibility and Stochastic Networks*, John Wiley and Sons, New York, 1979.

[Keme66] J. Kemeny, L. Snell, *Finite Markov Chains*, Van Nostrand, New York, 1966.

[Kers84] A. Kershenbaum, R. R. Boorstyn, "Evaluation of Throughput in Multihop Packet Radio Networks with Complex Topologies," *Proc. INFOCOM'84*, San Francisco, California, Apr. 1984.

[Klei64] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964. Reprinted by Dover Publications, 1972.

[Klei75a] L. Kleinrock, S. S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," *IEEE Trans. Comm.*, vol. COM-23, no. 4, Apr. 1975.

[Klei75b] L. Kleinrock, F. A. Tobagi, "Packet Switching in Radio Channels: Part I – Carrier Sense Multiple-Access Modes and their Throughput-Delay Characteristics," *IEEE Trans. Comm.*, vol. COM-23, no. 12, pp. 1400-1416, Dec. 1975.

[Klei76] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, John Wiley and Sons, New York, 1976.

[Klei78] L. Kleinrock, J. A. Silvester, "Optimum Transmission Radii for Packet Radio Networks, or why Six is a Magic Number," *Conf. Records Nat. Telecomm. Conf.*, pp. 4.3.1–4.3.5, Dec. 1978.

[Lam75] S. S. Lam, L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," *IEEE Trans. Comm.*, vol. COM-23, no. 9, pp. 891–904, Sept. 1975.

[Lawl76] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[Lee82] I. Lee, J. A. Silvester, "An Iterative Scheme for Performance Modeling of Slotted ALOHA Packet Radio Networks," *Proc. ICC'82*, pp. 6G.1.1–6G.1.5, June 1982.

[Luen73] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, 1973.

[Magl83] B. Maglaris, R. Boorstyn, A. Kershenbaum, "Extensions to the Analysis of Multihop Packet Radio Networks," *Proc. INFOCOM'83*, San Diego, California, May 1980.

[Nain85] P. Nain, "Analysis of a two-node ALOHA-network with infinite capacity buffers," *Proc. Int. Seminar Comp. Networking and Perf. Eval.*, pp. 2.2.1–2.2.15, Tokyo, 1985.

[Nels82] R. Nelson, "Channel Access Protocols for Multip-Hop Broadcast Packet Radio Networks," Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, *Engineering Report* UCLA–ENG–8118, CSD–810730, Jul. 1981.

[Nels84] R. Nelson, L. Kleinrock, "The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture," *IEEE Trans. Comm.*, vol. COM-32, no. 6, pp. 684–694, Jan. 1984.

[Nels85a] R. Nelson, L. Kleinrock, "Rude CSMA: A Multihop Channel Access Protocol," *IEEE Trans. Comm.*, vol. COM-33, no. 8, pp. 785-791, Aug. 1985.

[Nels85b] R. Nelson, L. Kleinrock, "Spatial-TDMA: A Collision-Free Multihop Channel Access Protocol," *IEEE Trans. Comm.*, vol. COM-33, no. 9, pp. 934-944, Sept. 1985.

[Pick82] R. L. Pickholtz, D. L. Schilling, L. B. Milstein, "Theory of Spread-Spectrum Communications - A Tutorial," *IEEE Trans. Comm.*, vol. COM-30, no. 5, pp. 855–885, May 1982.

[Purs77a] M. B. Pursley, "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication—Part I: System Analysis," *IEEE Trans. Comm.*, vol. COM-25, no. 8, pp. 795–799, Aug. 1977.

[Purs77b] M. B. Pursley, D. V. Sarwate, "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication—Part II: Code Sequence Analysis," *IEEE Trans. Comm.*, vol. COM-25, no. 8, pp. 800–803, Aug. 1977.

[Purs82a] M. B. Pursley, D. V. Sarwate, W. E. Stark, "Error Probability for Direct-Sequence Spread-Spectrum Multiple-Access Communication—Part I: Upper and Lower Bounds," *IEEE Trans. Comm.*, vol. COM-30, no. 5, pp. 975–984, May 1982.

[Purs83] M. Pursley, "Throughput of Frequency-Hopped Spread-Spectrum Communications for Packet Radio Networks," *Proc. 1983 CISS*, Johns Hopkins Univ., Mar. 1983.

[Rayc81] D. Raychauduri, "Performance Analysis of Random Access Packet-Switched Code Division Multiple Access Systems," *IEEE Trans. Comm.*, vol. COM-29, no. 6, pp. 895–901, Jun. 1981.

[Robe72] L. G. Roberts, "ALOHA Packet System with and without Slots and Capture," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.

[Ross70] S. Ross *Applied Probability Models with Optimization Applications*, Holden-Day, S. Francisco, 1970.

[Saad81] T. N. Saadawi, A. Ephremides, "Analysis, Stability, and Optimization of Slotted ALOHA with a Finite Number of Buffered Users," *IEEE Trans. Automatic Control,* vol. AC-26, no. 3, pp. 680–689, 1981.

[Scha77] R. Schassberger, "Insensitivity of Steady-State Distributions of Generalized Semi-Markov Processes. Part I," *Ann. Prob.,* vol. 5, no. 1, pp. 97–99, 1977.

[Scha78a] R. Schassberger, "Insensitivity of Steady-State Distributions of Generalized Semi-Markov Processes. Part II," *Ann. Prob.*, vol. 6, no. 1, pp. 85–93, 1978.

[Scha78b] R. Schassberger, "Insensitivity of Steady-State Distributions of Generalized Semi-Markov Processes with Speeds," *Adv. Appl. Prob.*, vol. 10, pp. 836–851, 1978.

[Shan79] K. S. Shanmugan, *Digital and Analog Communication Systems,* John Wiley and Sons, New York, 1979.

[Sidi83a] M. Sidi, A. Segall, "Two Interfering Queues in Packet-Radio Networks," *IEEE Trans. Comm.,* vol. COM-31, no. 1, Jan. 1983.

[Sidi83b] M. Sidi, A. Segall, "A Three-Node Packet-Radio Network," *Proceedings INFOCOM'83,* pp. 222–228, April, 1983.

[Silv80] J. A. Silvester, "On the Spatial Capacity of Packet Radio Networks," Computer Science Department, University of California, Los Angeles, *Eng. Report* UCLA–ENG–8021, May 1980.

[Silv82] J. A. Silvester, I. Lee, "Performance Modeling of Buffered CSMA – An Iterative Approach," *Conf. Record GLOBECOM'82,* pp. F1.6.1–F1.6.5, Miami, Nov. 1982.

[Sous85] E. Sousa, J. A. Silvester, "Determination of Optimum Transmission Ranges in a Multi-Hop Spread Spectrum Network," *Proceedings MILCOM'85*, pp. 25.3.1–25.3.5, Boston, Oct. 1985

[Stor84] J. S. Storey, F. A. Tobagi, "Improvements in Throughput of a CDMA Packet Radio Network Due to a Channel Load Sense Access Protocol," *SURAN Temporary Note* #20, Computer Systems Laboratory, Stanford University, Stanford, California, Jul. 1984.

[Stor85] J. S. Storey, F. A. Tobagi, "Throughput Performance of a Direct Sequence CDMA Packet Radio Network," *SEL Technical Report* no. 85–277, Computer Systems Laboratory, Stanford University, Stanford, California, Jun. 1985.

[Taka84] H. Takagi, L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.*, vol. COM-32, no. 3, pp. 246–257, Mar. 1984.

[Taka85] H. Takagi, L. Kleinrock, "Throughput-Delay Characteristics of Some Slotted-ALOHA Multihop Packet Radio Networks," *IEEE Trans. Comm.*, vol. COM-33, no. 11, pp. 1200-1207, Nov. 1985.

[Toba75] F. A. Tobagi, L. Kleinrock, "Packet Switching in Radio Broadcast Channels: Part II – The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy Tone Solution," *IEEE Trans. Comm.*, vol. COM-23, no. 12, pp. 1417-1433, Dec. 1975.

[Toba76] F. A. Tobagi, L. Kleinrock, "Packet Switching in Radio Broadcast Channels: Part III – Polling and (Dynamic) Split-Channel Reservation Multiple Access," *IEEE Trans. Comm.*, vol. COM-24, no. 8, pp. 832–845, Aug. 1976.

[Toba77] F. A. Tobagi, L. Kleinrock, "Packet Switching in Radio Broadcast Channels: Part IV – Stability Considerations and Dynamic Control in Carrier Sense Multiple Access," *IEEE Trans. Comm.*, vol. COM-24, no. 10, pp. 1103–1119, Oct. 1977.

[Toba80a] F. A. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network – Part I: Slotted ALOHA, *IEEE Trans. Comm.*, vol. COM-28, no. 2, pp. 196-207, Feb. 1980.

[Toba80b] F. A. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network – Part II: Carrier Sense Multiple Access," *IEEE Trans. Comm.*, vol. COM-28, no. 2, pp. 208–216, Feb. 1980.

[Toba83] F. A. Tobagi, J. M. Brázio, "Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes," *Proc. INFOCOM'83*, San Diego, California, Apr. 1983.

[Toba84] F. A. Tobagi, J. S. Storey, "Improvements in Throughput of a CDMA Packet Radio Network due to a Channel Load Sense Access Protocol," *Proc. 22nd Annual Allerton Conference on Commumication, Control and Computing,* Urbana, Illinois, Oct. 1984.

[Toba85] F. A. Tobagi, D. H. Shur, "Performance Evaluation of Channel Access Schemes in Multihop Paclet Radio Networks with Regular Structure by Simulation," *CSL Technical Report* no. 85-278, Stanford University, June 1985.

[Yao77] K. Yao, "Error Probability of Asynchronous Spread Spectrum Multiple Access Communication Systems", *IEEE Trans. Comm.*, vol. COM-25, no. 8, pp. 803–809, August 1977.

[Yemi80] Y. Yemini, "On Channel Sharing in Discrete-Time, Multi-Access Broadcast Communication," Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, *UCLA Tech. Report* UCLA–ENG–8061, Sept. 1980.

[Yemi83] Y. Yemini, "A Statistical Mechanics of Distributed Resource Mechanism Sharing," *Proc. INFOCOM'83,* San Diego, California, Apr. 1983.